



Technical Standard

1

2

Commands and Utilities, Issue 6

3

The Open Group

4

The Institute of Electrical and Electronics Engineers, Inc.



5

6 © 2000, *The Open Group*

7 © 2000, *The Institute of Electrical and Electronics Engineers, Inc.*

8 All rights reserved.

9 Except as permitted below, no part of this publication may be reproduced, stored in a retrieval
10 system, or transmitted, in any form or by any means, electronic, mechanical, photocopying,
11 recording or otherwise, without the prior permission of the copyright owners.

This is an unapproved draft, subject to change. Permission is hereby granted for Austin Group participants to reproduce IEEE Std. 1003.1-200x for purposes of IEEE, The Open Group, and JTC1 standardization activities.

Other entities seeking permission to reproduce IEEE Std. 1003.1-200x for standardization purposes or activities must contact the copyright owners for an appropriate license. Use of information contained within this unapproved draft is at your own risk.

Portions of IEEE Std. 1003.1-200x are derived with permission from copyrighted material owned by Hewlett-Packard Company, International Business Machines Corporation, Novell Inc., The Open Software Foundation, and Sun Microsystems, Inc.

12 Technical Standard

13 Commands and Utilities, Issue 6

14 Document Number:

15 Published in the U.K. by The Open Group, 2000.

16 See <http://www.opengroup.org/austin/bugreport.html> for instructions on commenting on this
17 unapproved draft document.

Contents

19	Chapter 1	Introduction.....	1
20	1.1	Scope.....	1
21	1.2	Conformance	3
22	1.3	Normative References	4
23	1.4	Changes from Issue 4.....	5
24	1.4.1	Changes from Issue 4 to Issue 4, Version 2.....	5
25	1.4.2	Changes from Issue 4, Version 2 to Issue 5.....	5
26	1.4.3	Changes from Issue 5 to Issue 6.....	5
27	1.5	Terminology.....	7
28	1.6	Definitions.....	9
29	1.7	Relationship to Other Documents.....	10
30	1.7.1	The System Interfaces volume of IEEEStd.1003.1-200x.....	10
31	1.7.1.1	Process Attributes.....	10
32	1.7.1.2	Concurrent Execution of Processes.....	10
33	1.7.1.3	File Access Permissions.....	10
34	1.7.1.4	File Read, Write, and Creation.....	11
35	1.7.1.5	File Removal	12
36	1.7.1.6	File Time Values.....	12
37	1.7.1.7	File Contents.....	12
38	1.7.1.8	Path Name Resolution.....	13
39	1.7.1.9	Changing the Current Working Directory.....	13
40	1.7.1.10	Establish the Locale	13
41	1.7.1.11	Actions Equivalent to Functions.....	13
42	1.8	Portability	14
43	1.8.1	Codes.....	14
44	1.9	Utility Limits.....	21
45	1.10	Grammar Conventions.....	24
46	1.11	Utility Description Defaults.....	25
47	1.12	Considerations for Utilities in Support of Files of Arbitrary Size	33
48	Chapter 2	Shell Command Language	35
49	2.1	Shell Introduction	35
50	2.2	Quoting.....	36
51	2.2.1	Escape Character (Backslash).....	36
52	2.2.2	Single-Quotes.....	36
53	2.2.3	Double-Quotes	36
54	2.3	Token Recognition.....	39
55	2.3.1	Alias Substitution	40
56	2.4	Reserved Words	42
57	2.5	Parameters and Variables.....	43
58	2.5.1	Positional Parameters.....	43
59	2.5.2	Special Parameters.....	43

60	2.5.3	Shell Variables.....	45
61	2.6	Word Expansions.....	49
62	2.6.1	Tilde Expansion.....	50
63	2.6.2	Parameter Expansion	51
64	2.6.3	Command Substitution	54
65	2.6.4	Arithmetic Expansion	56
66	2.6.5	Field Splitting.....	58
67	2.6.6	Path Name Expansion.....	59
68	2.6.7	Quote Removal.....	59
69	2.7	Redirection	60
70	2.7.1	Redirecting Input	61
71	2.7.2	Redirecting Output.....	61
72	2.7.3	Appending Redirected Output	61
73	2.7.4	Here-Document.....	61
74	2.7.5	Duplicating an Input File Descriptor	62
75	2.7.6	Duplicating an Output File Descriptor	62
76	2.7.7	Open File Descriptors for Reading and Writing.....	63
77	2.8	Exit Status and Errors	65
78	2.8.1	Consequences of Shell Errors	65
79	2.8.2	Exit Status for Commands.....	65
80	2.9	Shell Commands	67
81	2.9.1	Simple Commands	67
82	2.9.1.1	Command Search and Execution	69
83	2.9.2	Pipelines.....	72
84	2.9.3	Lists.....	73
85	2.9.3.1	Asynchronous Lists	74
86	2.9.3.2	Sequential Lists.....	75
87	2.9.3.3	AND Lists	75
88	2.9.3.4	OR Lists.....	75
89	2.9.4	Compound Commands.....	75
90	2.9.4.1	Grouping Commands	75
91	2.9.4.2	For Loop.....	76
92	2.9.4.3	Case Conditional Construct	77
93	2.9.4.4	If Conditional Construct	78
94	2.9.4.5	While Loop.....	78
95	2.9.4.6	Until Loop.....	79
96	2.9.5	Function Definition Command.....	79
97	2.10	Shell Grammar.....	82
98	2.10.1	Shell Grammar Lexical Conventions.....	82
99	2.10.2	Shell Grammar Rules	82
100	2.11	Signals and Error Handling	89
101	2.12	Shell Execution Environment.....	90
102	2.13	Pattern Matching Notation	92
103	2.13.1	Patterns Matching a Single Character	92
104	2.13.2	Patterns Matching Multiple Characters	93
105	2.13.3	Patterns Used for File Name Expansion	94
106	2.14	Special Built-In Utilities.....	96
107		<i>break</i>	97

108		<i>colon</i>	99
109		<i>continue</i>	101
110		<i>dot</i>	103
111		<i>eval</i>	105
112		<i>exec</i>	107
113		<i>exit</i>	109
114		<i>export</i>	111
115		<i>readonly</i>	113
116		<i>return</i>	115
117		<i>set</i>	117
118		<i>shift</i>	123
119		<i>times</i>	125
120		<i>trap</i>	127
121		<i>unset</i>	130
122	Chapter 3	Batch Environment Services and Utilities	133
123	3.1	General Concepts	133
124	3.1.1	Batch Client-Server Interaction	133
125	3.1.2	Batch Queues	134
126	3.1.3	Batch Job Creation	134
127	3.1.4	Batch Job Tracking	135
128	3.1.5	Batch Job Routing	135
129	3.1.6	Batch Job Execution	135
130	3.1.7	Batch Job Exit	136
131	3.1.8	Batch Job Abort	136
132	3.1.9	Batch Job States	136
133	3.1.10	Batch Authorization	136
134	3.1.11	Batch Administration	137
135	3.1.12	Batch Notification	137
136	3.2	Batch Services	137
137	3.2.1	Batch Job States	138
138	3.2.2	Deferred Batch Services	139
139	3.2.2.1	Batch Job Execution	139
140	3.2.2.2	Batch Job Routing	146
141	3.2.2.3	Batch Job Exit	146
142	3.2.2.4	Batch Server Restart	147
143	3.2.2.5	Batch Job Abort	147
144	3.2.3	Requested Batch Services	148
145	3.2.3.1	Delete Batch Job Request	148
146	3.2.3.2	Hold Batch Job Request	149
147	3.2.3.3	Batch Job Message Request	149
148	3.2.3.4	Batch Job Status Request	150
149	3.2.3.5	Locate Batch Job Request	150
150	3.2.3.6	Modify Batch Job Request	150
151	3.2.3.7	Move Batch Job Request	151
152	3.2.3.8	Queue Batch Job Request	151
153	3.2.3.9	Batch Queue Status Request	152
154	3.2.3.10	Release Batch Job Request	153

155	3.2.3.11	Rerun Batch Job Request	153
156	3.2.3.12	Select Batch Jobs Request	154
157	3.2.3.13	Server Shutdown Request	154
158	3.2.3.14	Server Status Request	154
159	3.2.3.15	Signal Batch Job Request	155
160	3.2.3.16	Track Batch Job Request	155
161	3.3	Common Behavior for Batch Environment Utilities	156
162	3.3.1	Batch Job Identifier	156
163	3.3.2	Destination	157
164	3.3.3	Multiple Keyword-Value Pairs	157
165	Chapter 4	Utilities	159
166		<i>admin</i>	160
167		<i>alias</i>	165
168		<i>ar</i>	168
169		<i>asa</i>	175
170		<i>at</i>	178
171		<i>awk</i>	187
172		<i>basename</i>	222
173		<i>batch</i>	225
174		<i>bc</i>	228
175		<i>bg</i>	243
176		<i>c89</i>	246
177		<i>cal</i>	257
178		<i>cat</i>	259
179		<i>cd</i>	263
180		<i>cflow</i>	267
181		<i>chgrp</i>	270
182		<i>chmod</i>	273
183		<i>chown</i>	279
184		<i>cksum</i>	283
185		<i>cmp</i>	288
186		<i>comm</i>	291
187		<i>command</i>	294
188		<i>compress</i>	299
189		<i>cp</i>	302
190		<i>crontab</i>	310
191		<i>csplit</i>	314
192		<i>ctags</i>	318
193		<i>cut</i>	323
194		<i>cxref</i>	327
195		<i>date</i>	330
196		<i>dd</i>	337
197		<i>delta</i>	344
198		<i>df</i>	348
199		<i>diff</i>	352
200		<i>dirname</i>	359
201		<i>du</i>	362

202	<i>echo</i>	366
203	<i>ed</i>	369
204	<i>env</i>	385
205	<i>ex</i>	388
206	<i>expand</i>	460
207	<i>expr</i>	463
208	<i>false</i>	468
209	<i>fc</i>	470
210	<i>fg</i>	476
211	<i>file</i>	479
212	<i>find</i>	486
213	<i>fold</i>	494
214	<i>fort77</i>	497
215	<i>fuser</i>	503
216	<i>gencat</i>	506
217	<i>get</i>	510
218	<i>getconf</i>	517
219	<i>getopts</i>	523
220	<i>grep</i>	528
221	<i>hash</i>	533
222	<i>head</i>	536
223	<i>iconv</i>	539
224	<i>id</i>	542
225	<i>ipcrm</i>	546
226	<i>ipcs</i>	548
227	<i>jobs</i>	555
228	<i>join</i>	559
229	<i>kill</i>	564
230	<i>lex</i>	568
231	<i>link</i>	580
232	<i>ln</i>	582
233	<i>locale</i>	586
234	<i>localedef</i>	591
235	<i>logger</i>	595
236	<i>logname</i>	597
237	<i>lp</i>	599
238	<i>ls</i>	604
239	<i>m4</i>	612
240	<i>mailx</i>	619
241	<i>make</i>	643
242	<i>man</i>	664
243	<i>mesg</i>	668
244	<i>mkdir</i>	671
245	<i>mkfifo</i>	674
246	<i>more</i>	677
247	<i>mv</i>	689
248	<i>newgrp</i>	694
249	<i>nice</i>	698

250	<i>nl</i>	702
251	<i>nm</i>	706
252	<i>nohup</i>	711
253	<i>od</i>	714
254	<i>paste</i>	722
255	<i>patch</i>	726
256	<i>pathchk</i>	732
257	<i>pax</i>	737
258	<i>pr</i>	772
259	<i>printf</i>	777
260	<i>prs</i>	782
261	<i>ps</i>	787
262	<i>pwd</i>	794
263	<i>qalter</i>	796
264	<i>qdel</i>	805
265	<i>qhold</i>	808
266	<i>qmove</i>	811
267	<i>qmsg</i>	814
268	<i>qrerun</i>	817
269	<i>qrls</i>	819
270	<i>qselect</i>	823
271	<i>qsig</i>	832
272	<i>qstat</i>	835
273	<i>qsub</i>	840
274	<i>read</i>	853
275	<i>renice</i>	856
276	<i>rm</i>	860
277	<i>rmdel</i>	865
278	<i>rmdir</i>	868
279	<i>sact</i>	871
280	<i>sccs</i>	874
281	<i>sed</i>	879
282	<i>sh</i>	888
283	<i>sleep</i>	903
284	<i>sort</i>	906
285	<i>split</i>	912
286	<i>strings</i>	916
287	<i>strip</i>	919
288	<i>stty</i>	921
289	<i>tabs</i>	931
290	<i>tail</i>	935
291	<i>talk</i>	939
292	<i>tee</i>	943
293	<i>test</i>	946
294	<i>time</i>	955
295	<i>touch</i>	959
296	<i>tput</i>	963
297	<i>tr</i>	966

298		<i>true</i>	972
299		<i>tsort</i>	974
300		<i>tty</i>	977
301		<i>type</i>	979
302		<i>ulimit</i>	981
303		<i>umask</i>	984
304		<i>unalias</i>	988
305		<i>uname</i>	990
306		<i>uncompress</i>	993
307		<i>unexpand</i>	996
308		<i>unget</i>	999
309		<i>uniq</i>	1002
310		<i>unlink</i>	1006
311		<i>uucp</i>	1008
312		<i>uudecode</i>	1012
313		<i>uuencode</i>	1015
314		<i>uustat</i>	1020
315		<i>uux</i>	1023
316		<i>val</i>	1027
317		<i>vi</i>	1030
318		<i>wait</i>	1084
319		<i>wc</i>	1088
320		<i>what</i>	1091
321		<i>who</i>	1094
322		<i>write</i>	1098
323		<i>xargs</i>	1101
324		<i>yacc</i>	1108
325		<i>zcat</i>	1124
326	Chapter 5	Portability Considerations (Informative)	1127
327	5.1	User Requirements	1127
328	5.1.1	Command Language	1128
329	5.1.2	Interactive Facilities	1128
330	5.1.3	Accomplish Multiple Tasks Simultaneously	1128
331	5.1.4	Complex Data Manipulation	1128
332	5.1.5	File Hierarchy Manipulation	1128
333	5.1.6	Locale Configuration	1128
334	5.1.7	Inter-User Communication	1129
335	5.1.8	System Environment	1129
336	5.1.9	Printing	1129
337	5.1.10	Software Development	1129
338	5.2	Portability Capabilities	1129
339	5.2.1	Interactive Facilities	1130
340	5.2.2	Accomplish Multiple Tasks Simultaneously	1130
341	5.2.3	Complex Data Manipulation	1131
342	5.2.4	File Hierarchy Manipulation	1131
343	5.2.5	Locale Configuration	1131
344	5.2.6	Inter-User Communication	1131

345	5.2.7	System Environment.....	1132
346	5.2.8	Printing.....	1132
347	5.2.9	Software Development.....	1132
348	5.3	Profiling Considerations.....	1132
349	5.3.1	Configuration Options.....	1133
350	5.3.2	Configurable Limits	1134
351	5.3.3	Optional Behavior.....	1134
352		Index.....	1135
353		List of Figures	
354	4-1	pax Format Archive Example.....	749
355		List of Tables	
356	1-1	Utility Limit Minimum Values.....	21
357	1-2	Symbolic Utility Limits.....	22
358	3-1	Batch Utilities.....	134
359	3-2	Environment Variable Summary.....	137
360	3-3	Next State Table	140
361	3-4	Results/Output Table	141
362	3-5	Batch Services Summary	148
363	4-1	Expressions in Decreasing Precedence in <i>awk</i>	190
364	4-2	Escape Sequences in <i>awk</i>	195
365	4-3	Operators in <i>bc</i>	233
366	4-4	Programming Environments: Type Sizes	250
367	4-5	Programming Environments: <i>c89</i> and <i>cc</i> Arguments.....	251
368	4-6	ASCII to EBCDIC Conversion.....	340
369	4-7	ASCII to IBM EBCDIC Conversion.....	340
370	4-8	File Utility Output Strings	481
371	4-9	Table Size Declarations in <i>lex</i>	571
372	4-10	Escape Sequences in <i>lex</i>	573
373	4-11	ERE Precedence in <i>lex</i>	574
374	4-12	Named Characters in <i>od</i>	717
375	4-13	ustar Header Block	753
376	4-14	ustar <i>mode</i> Field	755
377	4-15	Octet-Oriented <i>cpio</i> Archive Entry	757
378	4-16	Values for <i>cpio c_mode</i> Field	758
379	4-17	Variable Names and Default Headers in <i>ps</i>	791
380	4-18	Environment Variable Values (Utilities)	840
381	4-19	Control Character Names in <i>stty</i>	926
382	4-20	Circumflex Control Characters in <i>stty</i>	926
383	4-21	uuencode Base64 Values.....	1016
384	4-22	Internal Limits in <i>yacc</i>	1120
385			

This document is being jointly developed by the IEEE and The Open Group and is intended to become both IEEE Std. 1003.1-200x and an Open Group Technical Standard, making up the base volumes of the Single UNIX Specification, Version 3.

IEEE Std. 1003.1-200x

IEEE Std. 1003.1-200x defines the Portable Operating System Interface (POSIX) requirements and consists of the following volumes:

- System Interface Definitions
- Commands and Utilities (this volume)
- System Interfaces

This volume of IEEE Std. 1003.1-200x

The Commands and Utilities volume of IEEE Std. 1003.1-200x describes the commands and utilities offered to application programs on POSIX-conformant systems. Readers are expected to be familiar with the System Interface Definitions volume of IEEE Std. 1003.1-200x.

This volume of IEEE Std. 1003.1-200x is structured as follows:

- Chapter 1 explains the status of this volume of IEEE Std. 1003.1-200x and its relationship to other formal standards. It also describes the defaults used by the utility descriptions in Chapter 4.
- Chapter 2 describes the command language used in POSIX-conformant systems.
- Chapter 4 consists of reference pages for all utilities available on POSIX-conformant systems.

Comprehensive references are available in the index.

Typographical Conventions

The following typographical conventions are used throughout IEEE Std. 1003.1-200x:

- **Bold** font is used in text for options to commands, file names, keywords, type names, data structures, and their members.
- *Italic* strings are used for emphasis or to identify the first instance of a word requiring definition. Italics in text also denote:
 - Command operands, command option-arguments, or variable names; for example, substitutable argument prototypes
 - Environment variables, which are also shown in capitals
 - Utility names
 - External variables, such as *errno*
 - Functions; these are shown as follows: *name()*; names without parentheses are C external variables, C function family names, utility names, command operands, or command option-arguments.

- 421 • Normal font is used for the names of constants and literals.
- 422 • The notation `<file.h>` indicates a header.
- 423 • Names surrounded by braces, for example, {ARG_MAX}, represent symbolic limits or
- 424 configuration values which may be declared in appropriate headers by means of the C
- 425 `#define` construct.
- 426 • The notation [EABCD] is used to identify an error value EABCD.
- 427 • Syntax, code examples, and user input in interactive examples are shown in *fixed width*
- 428 font. Brackets shown in this font, [], are part of the syntax and do *not* indicate optional
- 429 items. In syntax the | symbol is used to separate alternatives, and ellipses (...) are used to
- 430 show that additional arguments are optional.
- 431 • **Bold fixed width** font is used to identify brackets that surround optional items in syntax,
- 432 [], and to identify system output in interactive examples.
- 433 • Variables within syntax statements are shown in *italic fixed width* font.
- 434 • Ranges of values are indicated with parentheses or brackets as follows:
- 435 — (a,b) means the range of all values from a to b, including neither a nor b.
- 436 — [a,b] means the range of all values from a to b, including a and b.
- 437 — [a,b) means the range of all values from a to b, including a, but not b.
- 438 — (a,b] means the range of all values from a to b, including b, but not a.
- 439 • Shading is used to identify extensions or warnings. |

Notes to Reviewers

This section with side shading will not appear in the final copy. - Ed.

This list will be revised at a later date.

The following information is given for the convenience of users of IEEE Std. 1003.1-200x and does not constitute an endorsement by The Open Group or IEEE of these products.

AT&T[®] is a registered trademark of AT&T in the U.S.A. and other countries.

Hewlett-Packard[®], HP[®], and HP-UX[®] are registered trademarks of Hewlett-Packard Company.

Motif[®], OSF/1[®], UNIX[®], and the “X Device” are registered trademarks and IT DialTone[™] and The Open Group[™] are trademarks of The Open Group in the U.S. and other countries.

POSIX[®] is a registered trademark of the Institute of Electrical and Electronic Engineers, Inc.

/usr/group[®] is a registered trademark of UniForum, the International Network of UNIX System Users.

Acknowledgements

453

454 **Notes to Reviewers**

455 *This section with side shading will not appear in the final copy. - Ed.*

456 This list will be revised at a later date.

457 The Open Group gratefully acknowledges:

- 458 • AT&T for permission to reproduce portions of its copyrighted System V Interface Definition
459 (SVID) and material from the UNIX System V Release 2.0 documentation.
- 460 • The ANSI X3J11 Committees.
- 461 • The Large File Summit for their work in developing the set of changes to the X/Open Single
462 UNIX Specification to support large files.
- 463 • The following individuals for their valuable contribution to the development of
464 IEEE Std. 1003.1-200x:

465 TBD

Normative References

The following standards contain provisions which, through references in IEEE Std. 1003.1-200x, constitute provisions of IEEE Std. 1003.1-200x. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this volume of IEEE Std. 1003.1-200x are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards.

ISO C

ISO/IEC 9899:1990: Programming Languages — C, including Amendment 1:1995 (E), C Integrity (Multibyte Support Extensions (MSE) for ISO C).

ISO/IEC 646:1991

ISO/IEC 646:1991, Information Technology — ISO 7-Bit Coded Character Set for Information Interchange.

The reference version of the standard contains 95 graphic characters, which are identical to the graphic characters defined in the ASCII coded character set.

ISO/IEC 10646-1:1993

ISO/IEC 10646-1:1993, Information Technology — Universal Multiple-Octet Coded Character Set (UCS) — Part 1: Architecture and Basic Multilingual Plane.

ISO/IEC 14519:1999

ISO/IEC 14519:1999, Information Technology — POSIX Ada Language Interfaces — Binding for System Application Program Interface (API) — Realtime Extensions.

ISO POSIX-2

ISO/IEC 9945-2:1993, Information Technology — Portable Operating System Interface (POSIX) — Part 2: Shell and Utilities (identical to IEEE Std. 1003.2-1992 as amended by IEEE Std. 1003.2a-1992).

Informative References

The following documents are referenced in IEEE Std. 1003.1-200x:

1984 /usr/group Standard

/usr/group Standards Committee, Santa Clara, CA, UniForum 1984.

Almasi and Gottlieb

George S. Almasi and Allan Gottlieb, *Highly Parallel Computing*, The Benjamin/Cummings Publishing Company, Inc., 1989, ISBN: 0-8053-0177-1.

ANS X3.9-1978

(Reaffirmed 1989) Programming Language FORTRAN.

ANSI C

American National Standard for Information Systems: Standard X3.159-1989, Programming Language C.

Brawer

Steven Brawer, *Introduction to Parallel Programming*, Academic Press, 1989, ISBN: 0-12-128470-0.

507	DeRemer, Frank, and Pennello Article
508	DeRemer, Frank, and Pennello, Thomas J., <i>Efficient Computation of LALR(1) Look-Ahead Sets</i> ,
509	SigPlan Notices, Volume 15, No. 8, August 1979.
510	Draft ANSI X3J11.1
511	IEEE Floating Point draft report of ANSI X3J11.1 (NCEG).
512	FIPS 151-1
513	Federal Information Procurement Standard (FIPS) 151-1.
514	FIPS 151-2
515	Federal Information Procurement Standards (FIPS) 151-2, Portable Operating System
516	Interface (POSIX)—Part 1: System Application Program Interface (API) [C Language].
517	HP-UX Manual
518	Hewlett-Packard HP-UX Release 9.0 Reference Manual, Third Edition, August 1992.
519	IEEE Std. 754: 1985
520	Standard for Binary Floating-Point Arithmetic.
521	IEEE Std. 854: 1987
522	Standard for Radix-Independent Floating-Point Arithmetic.
523	IETF RFC 791
524	Internet Protocol, Version 4 (IPv4), September 1981.
525	IETF RFC 819
526	The Domain Naming Convention for Internet User Applications, Z. Su, J. Postel, August
527	1982.
528	IETF RFC 822
529	Standard for the Format of ARPA Internet Text Messages, D.H. Crocker, August 1982.
530	IETF RFC 919
531	Broadcasting Internet Datagrams, J. Mogul, October 1984.
532	IETF RFC 920
533	Domain Requirements, J. Postel, J. Reynolds, October 1984.
534	IETF RFC 921
535	Domain Name System Implementation Schedule, J. Postel, October 1984.
536	IETF RFC 922
537	Broadcasting Internet Datagrams in the Presence of Subnets, J. Mogul, October 1984.
538	IETF RFC 1034
539	Domain Names — Concepts and Facilities, P. Mockapetris, November 1987.
540	IETF RFC 1035
541	Domain Names — Implementation and Specification, P. Mockapetris, November 1987.
542	IETF RFC 1123
543	Requirements for Internet Hosts — Application and Support, R. Braden, October 1989.
544	IETF RFC 1886
545	DNS Extensions to Support Internet Protocol, Version 6 (IPv6), C. Huitema, S. Thomson,
546	December 1995.
547	IETF RFC 2045
548	Multipurpose Internet Mail Extensions (MIME), Part 1: Format of Internet Message Bodies,
549	N. Freed, N. Borenstein, November 1996.

- 550 IETF RFC 2373
551 Internet Protocol, Version 6 (IPv6) Addressing Architecture, S. Deering, R. Hinden, July
552 1998.
- 553 IETF RFC 2460
554 Internet Protocol, Version 6 (IPv6), S. Deering, R. Hinden, December 1998.
- 555 Internationalisation Guide
556 Guide, July 1993, Internationalisation Guide, Version 2 (ISBN: 1-859120-02-4, G304),
557 published by The Open Group.
- 558 ISO 2375:1985
559 ISO 2375:1985, Data Processing — Procedure for Registration of Escape Sequences.
- 560 ISO 4217:1995
561 ISO 4217:1995, Codes for the Representation of Currencies and Funds.
- 562 ISO 8601:1988
563 ISO 8601:1988, Data Elements and Interchange Formats — Information Interchange —
564 Representation of Dates and Times.
- 565 ISO/IEC 1539:1990
566 ISO/IEC 1539:1990, Information Technology — Programming Languages — Fortran
567 (technically identical to ANSI standard X3.9-1978 [FORTRAN 77]).
- 568 ISO/IEC 4873:1991
569 ISO/IEC 4873:1991, Information Technology — ISO 8-Bit Code for Information Interchange
570 — Structure and Rules for Implementation.
- 571 ISO/IEC 6937:1994
572 ISO/IEC 6937:1994, Information Technology — Coded Character Set for Text
573 Communication — Latin Alphabet.
- 574 ISO 7-bit or 8-bit coded character set for text communication using public communication
575 networks, private communication networks, or interchange media, such as magnetic tapes
576 and discs.
- 577 ISO/IEC 8802-3:1996
578 ISO/IEC 8802-3:1996, Information Technology — Telecommunications and Information
579 Exchange Between Systems — Local and Metropolitan Area Networks — Specific
580 Requirements — Part 3: Carrier Sense Multiple Access with Collision Detection
581 (CSMA/CD) Access Method and Physical Layer Specifications.
- 582 ISO/IEC 8859-1:1998
583 ISO/IEC 8859-1:1998, Information Technology — 8-Bit Single-Byte Coded Graphic
584 Character Sets — Part 1: Latin Alphabet No. 1.
- 585 This standard character set comprises 191 graphic characters covering the requirements of
586 most of Western Europe.
- 587 ISO POSIX-1
588 ISO/IEC 9945-1:1996, Information Technology — Portable Operating System Interface
589 (POSIX) — Part 1: System Application Program Interface (API) [C Language] (identical to
590 ANSI/IEEE Std. 1003.1-1996). Incorporating ANSI/IEEE Stds. 1003.1-1990, 1003.1b-1993,
591 1003.1c-1995, and 1003.1i-1995.
- 592 Issue 1
593 X/Open Portability Guide, July 1985 (ISBN: 0-444-87839-4).

Issue 2

X/Open Portability Guide, January 1987:

- Volume 1: XVS Commands and Utilities (ISBN: 0-444-70174-5)
- Volume 2: XVS System Calls and Libraries (ISBN: 0-444-70175-3)

Issue 3

X/Open Specification, 1988, 1989, February 1992:

- Commands and Utilities, Issue 3 (ISBN: 1-872630-36-7, C211); this specification was formerly X/Open Portability Guide, Issue 3, Volume 1, January 1989, XSI Commands and Utilities (ISBN: 0-13-685835-X, XO/XPG/89/002)
- System Interfaces and Headers, Issue 3 (ISBN: 1-872630-37-5, C212); this specification was formerly X/Open Portability Guide, Issue 3, Volume 2, January 1989, XSI System Interface and Headers (ISBN: 0-13-685843-0, XO/XPG/89/003)
- Curses Interface, Issue 3, contained in Supplementary Definitions, Issue 3 (ISBN: 1-872630-38-3, C213), Chapters 9 to 14 inclusive; this specification was formerly X/Open Portability Guide, Issue 3, Volume 3, January 1989, XSI Supplementary Definitions (ISBN: 0-13-685850-3, XO/XPG/89/004)
- Headers Interface, Issue 3, contained in Supplementary Definitions, Issue 3 (ISBN: 1-872630-38-3, C213), Chapter 19, Cpio and Tar Headers; this specification was formerly X/Open Portability Guide Issue 3, Volume 3, January 1989, XSI Supplementary Definitions (ISBN: 0-13-685850-3, XO/XPG/89/004)

Issue 4

CAE Specification, July 1992, published by The Open Group:

- System Interface Definitions (XBD), Issue 4 (ISBN: 1-872630-46-4, C204)
- Commands and Utilities (XCU), Issue 4 (ISBN: 1-872630-48-0, C203)
- System Interfaces and Headers (XSH), Issue 4 (ISBN: 1-872630-47-2, C202)

Issue 4, Version 2

CAE Specification, August 1994, published by The Open Group:

- System Interface Definitions (XBD), Issue 4, Version 2 (ISBN: 1-85912-036-9, C434)
- Commands and Utilities (XCU), Issue 4, Version 2 (ISBN: 1-85912-034-2, C436)
- System Interfaces and Headers (XSH), Issue 4, Version 2 (ISBN: 1-85912-037-7, C435)

Issue 5

CAE Specification, January 1997, published by The Open Group:

- System Interface Definitions (XBD), Issue 5 (ISBN: 1-85912-186-1, C605)
- Commands and Utilities (XCU), Issue 5 (ISBN: 1-85912-191-8, C604)
- System Interfaces and Headers (XSH), Issue 5 (ISBN: 1-85912-181-0, C606)

Knuth Article

Knuth, Donald E., *On the Translation of Languages from Left to Right*, Information and Control, Volume 8, No. 6, October 1965.

KornShell

Bolsky, Morris I. and Korn, David G., *The New KornShell Command and Programming Language*, March 1995, Prentice Hall.

- 635 MSE working draft
636 Working draft of ISO/IEC 9899:1990/Add3:draft, Addendum 3 — Multibyte Support
637 Extensions (MSE) as documented in the ISO Working Paper SC22/WG14/N205 dated 31
638 March 1992.
- 639 POSIX.1: 1988
640 IEEE Std. 1003.1-1988, Standard for Information Technology — Portable Operating System
641 Interface (POSIX) — Part 1: System Application Program Interface (API) [C Language].
- 642 POSIX.1: 1990
643 IEEE Std. 1003.1-1990, Standard for Information Technology — Portable Operating System
644 Interface (POSIX) — Part 1: System Application Program Interface (API) [C Language].
- 645 POSIX.1d: 1999
646 IEEE Std. 1003.1d-1999, Standard for Information Technology — Portable Operating System
647 Interface (POSIX) — Part 1: System Application Program Interface (API) — Amendment
648 ????: Additional Realtime Extensions [C Language].
- 649 POSIX.1g: 2000
650 IEEE Std. 1003.1g-2000, Standard for Information Technology — Portable Operating System
651 Interface (POSIX) — Part 1: System Application Program Interface (API) — Amendment
652 ????: Protocol-Independent Interfaces (PII).
- 653 POSIX.1j: 2000
654 IEEE Std. 1003.1j-2000, Standard for Information Technology — Portable Operating System
655 Interface (POSIX) — Part 1: System Application Program Interface (API) — Amendment
656 ????: Advanced Realtime Extensions [C Language].
- 657 POSIX.2: 1992
658 IEEE Std. 1003.2-1992, Standard for Information Technology — Portable Operating System
659 Interface (POSIX) — Part 2: Shell and Utilities.
- 660 POSIX.2d:-1994
661 IEEE Std. 1003.2d: 1994, Standard for Information Technology — Portable Operating System
662 Interface (POSIX) — Part 2: Shell and Utilities — Amendment 1: Batch Environment.
- 663 Sarwate Article
664 Sarwate, Dilip V., *Computation of Cyclic Redundancy Checks via Table Lookup*, Communications
665 of the ACM, Volume 30, No. 8, August 1988.
- 666 SVID, Issue 1
667 American Telephone and Telegraph Company, System V Interface Definition (SVID), Issue
668 1; Morristown, NJ, UNIX Press, 1985.
- 669 SVID, Issue 2
670 American Telephone and Telegraph Company, System V Interface Definition (SVID), Issue
671 2; Morristown, NJ, UNIX Press, 1986.
- 672 SVID, Issue 3
673 American Telephone and Telegraph Company, System V Interface Definition (SVID), Issue
674 3; Morristown, NJ, UNIX Press, 1989.
- 675 The AWK Programming Language
676 Aho, Alfred V., Kernighan, Brian W., and Weinberger, Peter J., *The AWK Programming*
677 *Language*, Reading, MA, Addison-Wesley 1988.
- 678 XNS, Issue 4
679 CAE Specification, August 1994, Networking Services, Issue 4 (ISBN: 1-85912-049-0, C438),
680 published by The Open Group.

681	XNS, Issue 5	
682	CAE Specification, February 1997, Networking Services, Issue 5 (ISBN: 1-85912-165-9, C523),	
683	published by The Open Group.	
684	X/Open Curses, Issue 4, Version 2	
685	CAE Specification, May 1996, X/Open Curses, Issue 4, Version 2 (ISBN: 1-85912-171-3,	
686	C610), published by The Open Group.	
687	UTF-8	
688	ISO/IEC 10646-1: 1993/Amendment 2: 1996, UCS Transformation Format 8 (UTF-8).	
689	Yacc: Yet Another Compiler Compiler	
690	REFERENCE NEEDED.	
691	Parts of the following documents were used to create the base documents for	
692	IEEE Std. 1003.1-200x:	
693	AIX 3.2 Manual	
694	AIX Version 3.2 For RISC System/6000, Technical Reference: Base Operating System And	
695	Extensions, 1990, 1992 (Part No. SC23-2382-00).	
696	OSF/1	
697	OSF/1 Programmer's Reference, Release 1.2 (ISBN: 0-13-020579-6).	
698	OSF AES	
699	Application Environment Specification (AES) Operating System Programming Interfaces	
700	Volume, Revision A (ISBN: 0-13-043522-8).	
701	System V Release 2.0	
702	— UNIX System V Release 2.0 Programmer's Reference Manual (April 1984 - Issue 2).	
703	— UNIX System V Release 2.0 Programming Guide (April 1984 - Issue 2).	
704	System V Release 4.2	
705	Operating System API Reference, UNIX® SVR4.2 (1992) (ISBN: 0-13-017658-3).	
706		

Introduction

1.1 Scope

This volume of IEEE Std. 1003.1-200x defines a standard source code-level interface to command interpretation, or “shell”, services and common utility programs for application programs. These services and programs are complementary to those specified by the System Interfaces volume of IEEE Std. 1003.1-200x. The System Interface Definitions volume of IEEE Std. 1003.1-200x defines general terms, concepts, and interfaces used by this volume of IEEE Std. 1003.1-200x. When the User Portability Utilities option is included (see the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 2, Conformance), this volume of IEEE Std. 1003.1-200x has additional scope. The list of utilities and features is extended to provide a common interactive environment for system users and program developers.

This volume of IEEE Std. 1003.1-200x has been designed to be used by both application programmers and system implementors. When the User Portability Utilities option is included, it is also to be used by system users and program developers. However, it is intended to be a reference document and not a tutorial on the use of the services, the utilities, or the inter-relationships between the utilities.

The emphasis of this volume of IEEE Std. 1003.1-200x without the User Portability Utilities option is on the shell and utility functionality required by application programs (including “shell scripts”), and not on the direct interactive use of the shell command language or the utilities by humans. When the User Portability Utilities option is included, the emphasis is extended to support terminal users in a consistent manner across all conforming systems. There are three constraining factors that limit this user portability scope:

1. The *users* in this context are limited to the group of individuals who are familiar with the style of interaction characteristic of historically-derived systems based on one of the UNIX operating systems. Typical users would include program developers, engineers, or general-purpose time-sharing users.
2. The *environment* to be supported is a multi-user time-sharing system supporting character-oriented display terminals. Alternatively, it is a collection of single-user systems interconnected via local area networks or telephone lines, but with similar user interfaces. This volume of IEEE Std. 1003.1-200x does not include support that is tailored for bit-mapped or graphics display terminals, although it is expected that such terminals could emulate the character orientation required by this environment. When facilities require cursor addressability from the terminal hardware, this is specifically identified in this volume of IEEE Std. 1003.1-200x.
3. The facilities to be provided are based on the historical models of the following documents: the *System V Interface Definition*, the *BSD User Manual*, the *X/Open Portability Guide*, and documentation for the KornShell. Emphasis is placed on standardizing existing practice for existing users, with changes or additions limited to correcting deficiencies in the following areas:
 - a. Support for international character sets and other localization requirements, such as date formats, collation sequences, and so on
 - b. Reconciliation of differences between the historical implementations

c. Elimination of system or device dependencies

d. Corrections of features that could reduce system or user security/integrity

When the Batch Environment option is included, IEEE Std. 1003.1-200x has additional scope. The list of utilities and features is extended to provide users with a common method to submit work for deferred processing. This represents support for distributed batch environments, and is intended to support application and user portability and system interoperability in heterogeneous environments. Distributed batch environments include traditional batch processing and other generic queue processes.

This volume of IEEE Std. 1003.1-200x is based upon documentation and the knowledge of existing programs that assume an interface and architecture similar to that described by the System Interfaces volume of IEEE Std. 1003.1-200x. Any questions regarding the definition of terms or the semantics of an underlying concept should be referred to the System Interfaces volume of IEEE Std. 1003.1-200x.

56 **1.2 Conformance**

57 Conformance requirements for IEEE Std. 1003.1-200x are defined in the System Interface
58 Definitions volume of IEEE Std. 1003.1-200x, Chapter 2, Conformance.

1.3 Normative References

The following standards contain provisions which, through references in this text, constitute provisions of this volume of IEEE Std. 1003.1-200x. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this volume of IEEE Std. 1003.1-200x are encouraged to investigate the possibility of applying the most recent editions of the standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards.

Notes to Reviewers

This section with side shading will not appear in the final copy. - Ed.

The following list will be updated.

ISO/IEC 646

ISO/IEC 646: 1991, Information Processing — ISO 7-bit Coded Character Set for Information Interchange.¹

ISO 4217

ISO 4217: 1990, Codes for the Representation of Currencies and Funds.

ISO/IEC 4873

ISO/IEC 4873: 1991, Information Technology — ISO 8-bit Code for Information Interchange — Structure and Rules for Implementation.

ISO 8601

ISO 8601: 1988, Data Elements and Interchange Formats — Information Interchange — Representation of Dates and Times.

ISO 8859-1

ISO 8859-1: 1987, Information Processing — 8-bit Single-byte Coded Graphic Character Sets — Part 1: Latin Alphabet No. 1.

ISO 8859-2

ISO 8859-2: 1987, Information Processing — 8-bit Single-byte Coded Graphic Character Sets — Part 2: Latin Alphabet No. 2.

ISO/IEC 9899

ISO/IEC 9899: 1990, Programming Languages — C.

ISO/IEC 9945-1

ISO/IEC 9945-1: 200x, Information Technology — Portable Operating System Interface (POSIX) — Part 1: System Application Program Interface (API) [C Language] (identical to ANSI/IEEE Std 1003.1-200x).²

ANS X3.9-1978

(Reaffirmed 1989) American National Standard Programming Language FORTRAN.³

-
1. ISO/IEC documents can be obtained from the ISO office: 1 Rue de Varembe, Case Postale 56, CH-1211, Genève 20, Switzerland/Suisse
 2. This standard is available from the IEEE Service Center, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331, U.S.A. Tel: 1 (800) 678-IEEE or +1 (908) 981-1393.
 3. ANSI documents can be obtained from the Sales Department, American National Standards Institute, 1430 Broadway, New York, NY 10018, U.S.A.

1.4 Changes from Issue 4

Notes to Reviewers

This section with side shading will not appear in the final copy. - Ed.

The change history is subject to revision. The intention is to keep change history from Issue 4, and in the Issue 5 to Issue 6 change history to note changes from POSIX.2-1992 as well as Issue 5.

The following sections describe changes made to this volume of IEEE Std. 1003.1-200x since Issue 4. The CHANGE HISTORY section for each utility describes technical changes made to that utility since Issue 4. Changes made between Issue 2 and Issue 4 are not included.

1.4.1 Changes from Issue 4 to Issue 4, Version 2

The following list summarizes the major changes that were made in this volume of IEEE Std. 1003.1-200x from Issue 4 to Issue 4, Version 2:

- The X/Open UNIX extension was added, which specifies the common core utilities of 4.3 Berkeley Software Distribution (4.3 BSD), the OSF AES, and SVID Issue 3.

1.4.2 Changes from Issue 4, Version 2 to Issue 5

The following list summarizes the major changes that were made in this volume of IEEE Std. 1003.1-200x from Issue 4, Version 2 to Issue 5:

- Large File Summit (LFS) Extensions were added.
- Some utilities were updated to reflect changes for the POSIX Realtime Extension.
- Some utilities were updated to reflect changes for the POSIX Threads Extension.
- The LEGACY category of utilities was introduced as a replacement for the TO BE WITHDRAWN, WITHDRAWN, and Possibly Unsupportable categories.
- The following utilities were added:

fuser

ipcrm

ipcs

link

unlink

1.4.3 Changes from Issue 5 to Issue 6

The following list summarizes the major changes that were made in this volume of IEEE Std. 1003.1-200x from Issue 5 to Issue 6:

- This volume of IEEE Std. 1003.1-200x is extensively revised so it can be both an IEEE POSIX Standard and an Open Group Technical Standard.
- this volume of IEEE Std. 1003.1-200x is updated to mandate support of FIPS 151-2. The following changes were made:
 - Support is mandated for the capabilities associated with the following symbolic constants:

- 137 _POSIX_CHOWN_RESTRICTED
138 _POSIX_JOB_CONTROL
139 _POSIX_SAVED_IDS
- 140 — In the environment for the login shell, the environment variables *LOGNAME* and *HOME* |
141 shall be defined and have the properties described in the System Interface Definitions |
142 volume of IEEE Std. 1003.1-200x, Chapter 7, Locale. |
- 143 • this volume of IEEE Std. 1003.1-200x is updated to align with some features of the Single
144 UNIX Specification.
- 145 • A RATIONALE section is added to each reference page.
- 146 • A new chapter on Conformance is added.
- 147 • A new chapter on Portability Considerations is added.

1.5 Terminology

This section appears in the System Interface Definitions volume of IEEE Std. 1003.1-200x, but is repeated here for convenience:

For the purposes of IEEE Std. 1003.1-200x, the following terminology definitions apply:

can

Describes a permissible optional feature or behavior available to the user or application. The feature or behavior is mandatory for an implementation that conforms to IEEE Std. 1003.1-200x. An application can rely on the existence of the feature or behavior.

implementation-dependent

(Same meaning as *implementation-defined*.) Describes a value or behavior that is not defined by IEEE Std. 1003.1-200x but is selected by an implementor. The value or behavior may vary among implementations that conform to IEEE Std. 1003.1-200x. An application should not rely on the existence of the value or behavior. An application that relies on such a value or behavior cannot be assured to be portable across conforming implementations.

The implementor shall document such a value or behavior so that it can be used correctly by an application.

legacy

Describes a feature or behavior that is being retained for compatibility with older applications, but which has limitations which make it inappropriate for developing portable applications. New applications should use alternative means of obtaining equivalent functionality.

may

Describes a feature or behavior that is optional for an implementation that conforms to IEEE Std. 1003.1-200x. An application should not rely on the existence of the feature or behavior. An application that relies on such a feature or behavior cannot be assured to be portable across conforming implementations.

To avoid ambiguity, the opposite of *may* is expressed as *need not*, instead of *may not*.

shall

For an implementation that conforms to IEEE Std. 1003.1-200x, describes a feature or behavior that is mandatory. An application can rely on the existence of the feature or behavior.

For an application or user, describes a behavior that is mandatory.

should

For an implementation that conforms to IEEE Std. 1003.1-200x, describes a feature or behavior that is recommended but not mandatory. An application should not rely on the existence of the feature or behavior. An application that relies on such a feature or behavior cannot be assured to be portable across conforming implementations.

For an application, describes a feature or behavior that is recommended programming practice for optimum portability.

undefined

Describes the nature of a value or behavior not defined by IEEE Std. 1003.1-200x which results from use of an invalid program construct or invalid data input.

The value or behavior may vary among implementations that conform to IEEE Std. 1003.1-200x. An application should not rely on the existence or validity of the value or behavior. An application that relies on any particular value or behavior cannot be

193	assured to be portable across conforming implementations.	
194	unspecified	
195	Describes the nature of a value or behavior not specified by IEEE Std. 1003.1-200x which	
196	results from use of a valid program construct or valid data input.	
197	The value or behavior may vary among implementations that conform to	
198	IEEE Std. 1003.1-200x. An application should not rely on the existence or validity of the	
199	value or behavior. An application that relies on any particular value or behavior cannot be	
200	assured to be portable across conforming implementations.	
201	will	
202	Same meaning as <i>shall</i> ; <i>shall</i> is the preferred term.	

203 **1.6 Definitions**

204 Concepts and definitions are defined in the System Interface Definitions volume of
205 IEEE Std. 1003.1-200x.

1.7 Relationship to Other Documents

1.7.1 The System Interfaces volume of IEEE Std. 1003.1-200x

This subsection describes some of the features provided by the System Interfaces volume of IEEE Std. 1003.1-200x that are assumed to be globally available by all systems conforming to this volume of IEEE Std. 1003.1-200x. This subsection does not attempt to detail all of the features defined in the System Interfaces volume of IEEE Std. 1003.1-200x that are required by all of the utilities defined in this volume of IEEE Std. 1003.1-200x; the utility and function descriptions point out additional functionality required to provide the corresponding specific features needed by each.

The following subsections describe frequently used concepts. Many of these concepts are described in the System Interface Definitions volume of IEEE Std. 1003.1-200x. Utility and function description statements override these defaults when appropriate.

1.7.1.1 Process Attributes

The following process attributes, as described in the System Interfaces volume of IEEE Std. 1003.1-200x, are assumed to be supported for all processes in this volume of IEEE Std. 1003.1-200x:

Controlling Terminal	Real Group ID
Current Working Directory	Real User ID
Effective Group ID	Root Directory
Effective User ID	Saved Set-Group-ID
File Descriptors	Saved Set-User-ID
File Mode Creation Mask	Session Membership
Process Group ID	Supplementary Group IDs
Process ID	

A conforming implementation may include additional process attributes.

1.7.1.2 Concurrent Execution of Processes

The following functionality of the *fork()* function defined in the System Interfaces volume of IEEE Std. 1003.1-200x shall be available on all systems conforming to this volume of IEEE Std. 1003.1-200x:

1. Independent processes shall be capable of executing independently without either process terminating.
2. A process shall be able to create a new process with all of the attributes referenced in Section 1.7.1.1, determined according to the semantics of a call to the *fork()* function defined in the System Interfaces volume of IEEE Std. 1003.1-200x followed by a call in the child process to one of the *exec* functions defined in the System Interfaces volume of IEEE Std. 1003.1-200x.

1.7.1.3 File Access Permissions

The file access control mechanism described by the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 4.1, File Access Permissions applies to all files on an implementation conforming to this volume of IEEE Std. 1003.1-200x.

1.7.1.4 File Read, Write, and Creation

When a file is to be read or written, the file shall be opened with an access mode corresponding to the operation to be performed. If file access permissions deny access, the requested operation shall fail.

When a file that does not exist is created, the following features defined in the System Interfaces volume of IEEE Std. 1003.1-200x shall apply unless the utility or function description states otherwise:

1. The user ID of the file is set to the effective user ID of the calling process.
2. The group ID of the file is set to the effective group ID of the calling process or the group ID of the directory in which the file is being created.
3. If the file is a regular file, the permission bits of the file are set to:

$$S_IROTH \mid S_IWOTH \mid S_IRGRP \mid S_IWGRP \mid S_IRUSR \mid S_IWUSR$$
 (see the description of *File Modes* in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 13, Headers, <**sys/stat.h**>) except that the bits specified by the file mode creation mask of the process are cleared. If the file is a directory, the permission bits are set to:

$$S_IRWXU \mid S_IRWXG \mid S_IRWXO$$
 except that the bits specified by the file mode creation mask of the process are cleared.
4. The *st_atime*, *st_ctime*, and *st_mtime* fields of the file shall be updated as specified in the System Interfaces volume of IEEE Std. 1003.1-200x, Section 2.5, Standard I/O Streams.
5. If the file is a directory, it shall be an empty directory; otherwise, the file shall have length zero.
6. If the file is a symbolic link, the effect shall be undefined unless the {POSIX2_SYMLINKS} variable is in effect for the directory in which the symbolic link would be created.
7. Unless otherwise specified, the file created shall be a regular file.

When an attempt is made to create a file that already exists, the action shall depend on the file type:

1. For directories and FIFO special files, the attempt shall fail and the utility shall either continue with its operation or exit immediately with a non-zero status, depending on the description of the utility.
2. For regular files:
 - a. The user ID, group ID, and permission bits of the file shall not be changed.
 - b. The file shall be truncated to zero length.
 - c. The *st_ctime* and *st_mtime* fields shall be marked for update.
3. For other file types, the effect is implementation-dependent.

When a file is to be appended, the file shall be opened in a manner equivalent to using the O_APPEND flag, without the O_TRUNC flag, in the *open()* function defined in the System Interfaces volume of IEEE Std. 1003.1-200x.

284 1.7.1.5 File Removal

285 When a directory that is the root directory or current working directory of any process is
 286 removed, the effect is implementation-dependent. If file access permissions deny access, the
 287 requested operation fails. Otherwise, when a file is removed:

- 288 1. Its directory entry is removed from the file system.
- 289 2. The link count of the file is decremented.
- 290 3. If the file is an empty directory (see the System Interface Definitions volume of
 291 IEEE Std. 1003.1-200x, Section 3.145, Empty Directory):
 - 292 a. If no process has the directory open, the space occupied by the directory is freed and
 293 the directory is no longer accessible.
 - 294 b. If one or more processes have the directory open, the directory contents are
 295 preserved until all references to the file have been closed.
- 296 4. If the file is a directory that is not empty, the *st_ctime* field is marked for update.
- 297 5. If the file is not a directory:
 - 298 a. If the link count becomes zero:
 - 299 i. If no process has the file open, the space occupied by the file is freed and the
 300 file is no longer accessible.
 - 301 ii. If one or more processes have the file open, the file contents are preserved until
 302 all references to the file have been closed.
 - 303 b. If the link count is not reduced to zero, the *st_ctime* field is marked for update.
- 304 6. The *st_ctime* and *st_mtime* fields of the containing directory are marked for update.

305 1.7.1.6 File Time Values

306 All files shall have the three time values described by the System Interface Definitions volume of
 307 IEEE Std. 1003.1-200x, Section 4.3, File Times Update.

308 1.7.1.7 File Contents

309 When a reference is made to the contents of a file, *pathname*, this means the equivalent of all of
 310 the data placed in the space pointed to by *buf* when performing the *read()* function calls in the
 311 following operations defined in the System Interfaces volume of IEEE Std. 1003.1-200x:

```
312 while (read (fildes, buf, nbytes) > 0)
313     ;
```

314 If the file is indicated by a path name *pathname*, the file descriptor shall be determined by the
 315 equivalent of the following operation defined in the System Interfaces volume of
 316 IEEE Std. 1003.1-200x:

```
317 fildes = open (pathname, O_RDONLY);
```

318 The value of *nbytes* in the above sequence is unspecified; if the file is of a type where the data
 319 returned by *read()* would vary with different values, the value is one that results in the most
 320 data being returned.

321 If the *read()* function calls would return an error, it is unspecified whether the contents of the file
 322 are considered to include any data from offsets in the file beyond where the error would be
 323 returned.

324 1.7.1.8 Path Name Resolution

325 The path name resolution algorithm, described by the System Interface Definitions volume of
326 IEEE Std. 1003.1-200x, Section 4.5, Path Name Resolution, is used by implementations
327 conforming to this volume of IEEE Std. 1003.1-200x; see also the System Interface Definitions
328 volume of IEEE Std. 1003.1-200x, Section 3.171, File Hierarchy.

329 1.7.1.9 Changing the Current Working Directory

330 When the current working directory (see the System Interface Definitions volume of
331 IEEE Std. 1003.1-200x, Section 3.433, Working Directory) is to be changed, unless the utility or
332 function description states otherwise, the operation shall succeed unless a call to the *chdir()*
333 function defined in the System Interfaces volume of IEEE Std. 1003.1-200x would fail when
334 invoked with the new working directory path name as its argument.

335 1.7.1.10 Establish the Locale

336 The functionality of the *setlocale()* function defined in the System Interfaces volume of
337 IEEE Std. 1003.1-200x is assumed to be available on all systems conforming to this volume of
338 IEEE Std. 1003.1-200x; that is, utilities that require the capability of establishing an international
339 operating environment shall be permitted to set the specified category of the international
340 environment.

341 1.7.1.11 Actions Equivalent to Functions

342 Some utility descriptions specify that a utility performs actions equivalent to a function defined
343 in the System Interfaces volume of IEEE Std. 1003.1-200x. Such specifications require only that
344 the external effects be equivalent, not that any effect within the utility and visible only to the
345 utility be equivalent.

1.8 Portability

Some of the utilities in the Commands and Utilities volume of IEEE Std. 1003.1-200x and functions in the System Interfaces volume of IEEE Std. 1003.1-200x describe functionality that might not be fully portable to systems meeting the requirements for POSIX conformance (see the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 2, Conformance).

Where optional, enhanced, or reduced functionality is specified, the text is shaded and a code in the margin identifies the nature of the option, extension, or warning (see Section 1.8.1). For maximum portability, an application should avoid such functionality.

Unless the primary task of a utility is to produce textual material on its standard output, application developers should not rely on the format or content of any such material that may be produced. Where the primary task *is* to provide such material, but the output format is incompletely specified, the description is marked with the OF margin code and shading. Application developers are warned not to expect that the output of such an interface on one system is any guide to its behavior on another system.

1.8.1 Codes

Codes and their meanings are listed in the System Interface Definitions volume of IEEE Std. 1003.1-200x, but are repeated here for convenience:

ADV Advisory Information

The functionality described is optional. The functionality described is also an extension to the ISO C standard.

Where applicable, functions are marked with the ADV margin legend in the SYNOPSIS section. Where additional semantics apply to a function, the material is identified by use of the ADV margin legend.

AIO Asynchronous Input and Output

The functionality described is optional. The functionality described is also an extension to the ISO C standard.

Where applicable, functions are marked with the AIO margin legend in the SYNOPSIS section. Where additional semantics apply to a function, the material is identified by use of the AIO margin legend.

BAR Barriers

The functionality described is optional. The functionality described is also an extension to the ISO C standard.

Where applicable, functions are marked with the BAR margin legend in the SYNOPSIS section. Where additional semantics apply to a function, the material is identified by use of the BAR margin legend.

BE Batch Environment Services and Utilities

The functionality described is optional.

Where applicable, utilities are marked with the BE margin legend in the SYNOPSIS section. Where additional semantics apply to a utility, the material is identified by use of the BE margin legend.

CD C-Language Development Utilities

The functionality described is optional.

Where applicable, utilities are marked with the CD margin legend in the SYNOPSIS section. Where additional semantics apply to a utility, the material is identified by use of the CD margin

390		legend.
391	CPT	Process CPU-Time Clocks
392		The functionality described is optional. The functionality described is also an extension to the
393		ISO C standard.
394		Where applicable, functions are marked with the CPT margin legend in the SYNOPSIS section.
395		Where additional semantics apply to a function, the material is identified by use of the CPT
396		margin legend.
397	CS	Clock Selection
398		The functionality described is optional. The functionality described is also an extension to the
399		ISO C standard.
400		Where applicable, functions are marked with the CS margin legend in the SYNOPSIS section.
401		Where additional semantics apply to a function, the material is identified by use of the CS
402		margin legend.
403	CX	Extension to the ISO C standard
404		The functionality described is an extension to the ISO C standard. Application writers may
405		confidently make use of an extension as it is supported on all IEEE Std. 1003.1-200x conforming
406		systems.
407	FD	FORTTRAN Development Utilities
408		The functionality described is optional.
409		Where applicable, utilities are marked with the FD margin legend in the SYNOPSIS section.
410		Where additional semantics apply to a utility, the material is identified by use of the FD margin
411		legend.
412	FR	FORTTRAN Runtime Utilities
413		The functionality described is optional.
414		Where applicable, utilities are marked with the FR margin legend in the SYNOPSIS section.
415		Where additional semantics apply to a utility, the material is identified by use of the FR margin
416		legend.
417	FSC	File Synchronization
418		The functionality described is optional. The functionality described is also an extension to the
419		ISO C standard.
420		Where applicable, functions are marked with the FSC margin legend in the SYNOPSIS section.
421		Where additional semantics apply to a function, the material is identified by use of the FSC
422		margin legend.
423	IP6	IPV6
424		The functionality described is optional. The functionality described is also an extension to the
425		ISO C standard.
426		Where applicable, functions are marked with the IP6 margin legend in the SYNOPSIS section.
427		Where additional semantics apply to a function, the material is identified by use of the IP6
428		margin legend.
429	MAN	Mandatory in the Next Draft
430		This is an interim draft code used to aid reviewers during the development of
431		IEEE Std. 1003.1-200x. It denotes a feature that was previously an option or extension that is
432		being brought into the mandatory base functionality. This margin code will be removed from the
433		final draft.

434	MF	Memory Mapped Files
435		The functionality described is optional. The functionality described is also an extension to the
436		ISO C standard.
437		Where applicable, functions are marked with the MF margin legend in the SYNOPSIS section.
438		Where additional semantics apply to a function, the material is identified by use of the MF
439		margin legend.
440	ML	Process Memory Locking
441		The functionality described is optional. The functionality described is also an extension to the
442		ISO C standard.
443		Where applicable, functions are marked with the ML margin legend in the SYNOPSIS section.
444		Where additional semantics apply to a function, the material is identified by use of the ML
445		margin legend.
446	MLR	Range Memory Locking
447		The functionality described is optional. The functionality described is also an extension to the
448		ISO C standard.
449		Where applicable, functions are marked with the MLR margin legend in the SYNOPSIS section.
450		Where additional semantics apply to a function, the material is identified by use of the MLR
451		margin legend.
452	MON	Monotonic Clock
453		The functionality described is optional. The functionality described is also an extension to the
454		ISO C standard.
455		Where applicable, functions are marked with the MON margin legend in the SYNOPSIS section.
456		Where additional semantics apply to a function, the material is identified by use of the MON
457		margin legend.
458	MPR	Memory Protection
459		The functionality described is optional. The functionality described is also an extension to the
460		ISO C standard.
461		Where applicable, functions are marked with the MPR margin legend in the SYNOPSIS section.
462		Where additional semantics apply to a function, the material is identified by use of the MPR
463		margin legend.
464	MSG	Message Passing
465		The functionality described is optional. The functionality described is also an extension to the
466		ISO C standard.
467		Where applicable, functions are marked with the MSG margin legend in the SYNOPSIS section.
468		Where additional semantics apply to a function, the material is identified by use of the MSG
469		margin legend.
470	OB	Obsolescent
471		The functionality described may be withdrawn in a future version of this volume of
472		IEEE Std. 1003.1-200x. Strictly Conforming POSIX Applications and Strictly Conforming XSI
473		Applications shall not use obsolescent features.
474	OF	Output Format Incompletely Specified
475		The functionality described is an XSI extension. The format of the output produced by the utility
476		is not fully specified. It is therefore not possible to post-process this output in a consistent
477		fashion. Typical problems include unknown length of strings and unspecified field delimiters.

478	OH	Optional Header
479		In the SYNOPSIS section of some interfaces in the System Interfaces volume of
480		IEEE Std. 1003.1-200x an included header is marked as in the following example:
481	OH	<code>#include <sys/types.h></code>
482		<code>#include <grp.h></code>
483		<code>struct group *getgrnam(const char *name);</code>
484		This indicates that the marked header is not required on XSI-conformant systems.
485	PI	The Behavior Cannot be Guaranteed to be Consistent
486		The functionality described is an XSI extension. It is not possible to guarantee that the utility
487		behaves in the same way on all conformant systems. This is the case if it provides functionality
488		that is implementation-dependent. Options that are used to <i>select</i> alternative forms of
489		implementation-dependent behavior are not marked, as it is clear from their descriptions that
490		their use is inherently non-portable.
491	PIO	Prioritized Input and Output
492		The functionality described is optional. The functionality described is also an extension to the
493		ISO C standard.
494		Where applicable, functions are marked with the PIO margin legend in the SYNOPSIS section.
495		Where additional semantics apply to a function, the material is identified by use of the PIO
496		margin legend.
497	PS	Process Scheduling
498		The functionality described is optional. The functionality described is also an extension to the
499		ISO C standard.
500		Where applicable, functions are marked with the PS margin legend in the SYNOPSIS section.
501		Where additional semantics apply to a function, the material is identified by use of the PS
502		margin legend.
503	RTS	Realtime Signals Extension
504		The functionality described is optional. The functionality described is also an extension to the
505		ISO C standard.
506		Where applicable, functions are marked with the RTS margin legend in the SYNOPSIS section.
507		Where additional semantics apply to a function, the material is identified by use of the RTS
508		margin legend.
509	RWL	Reader/Writer Locks
510		The functionality described is optional. The functionality described is also an extension to the
511		ISO C standard.
512		Where applicable, functions are marked with the RWL margin legend in the SYNOPSIS section.
513		Where additional semantics apply to a function, the material is identified by use of the RWL
514		margin legend.
515	SD	Software Development Utilities
516		The functionality described is optional.
517		Where applicable, utilities are marked with the SD margin legend in the SYNOPSIS section.
518		Where additional semantics apply to a utility, the material is identified by use of the SD margin
519		legend.
520	SEM	Semaphores
521		The functionality described is optional. The functionality described is also an extension to the
522		ISO C standard.

523		Where applicable, functions are marked with the SEM margin legend in the SYNOPSIS section.
524		Where additional semantics apply to a function, the material is identified by use of the SEM
525		margin legend.
526	SHM	Shared Memory Objects
527		The functionality described is optional. The functionality described is also an extension to the
528		ISO C standard.
529		Where applicable, functions are marked with the SHM margin legend in the SYNOPSIS section.
530		Where additional semantics apply to a function, the material is identified by use of the SHM
531		margin legend.
532	SIO	Synchronized Input and Output
533		The functionality described is optional. The functionality described is also an extension to the
534		ISO C standard.
535		Where applicable, functions are marked with the SIO margin legend in the SYNOPSIS section.
536		Where additional semantics apply to a function, the material is identified by use of the SIO
537		margin legend.
538	SPI	Spin Locks
539		The functionality described is optional. The functionality described is also an extension to the
540		ISO C standard.
541		Where applicable, functions are marked with the SPI margin legend in the SYNOPSIS section.
542		Where additional semantics apply to a function, the material is identified by use of the SPI
543		margin legend.
544	SPN	Spawn
545		The functionality described is optional. The functionality described is also an extension to the
546		ISO C standard.
547		Where applicable, functions are marked with the SPN margin legend in the SYNOPSIS section.
548		Where additional semantics apply to a function, the material is identified by use of the SPN
549		margin legend.
550	SS	Process Sporadic Server
551		The functionality described is optional. The functionality described is also an extension to the
552		ISO C standard.
553		Where applicable, functions are marked with the SS margin legend in the SYNOPSIS section.
554		Where additional semantics apply to a function, the material is identified by use of the SS
555		margin legend.
556	TCT	Thread CPU-Time Clocks
557		The functionality described is optional. The functionality described is also an extension to the
558		ISO C standard.
559		Where applicable, functions are marked with the TCT margin legend in the SYNOPSIS section.
560		Where additional semantics apply to a function, the material is identified by use of the TCT
561		margin legend.
562	THR	Threads
563		The functionality described is optional. The functionality described is also an extension to the
564		ISO C standard.
565		Where applicable, functions are marked with the THR margin legend in the SYNOPSIS section.
566		Where additional semantics apply to a function, the material is identified by use of the THR
567		margin legend.

568	TMO	Timeouts
569		The functionality described is optional. The functionality described is also an extension to the
570		ISO C standard.
571		Where applicable, functions are marked with the TMO margin legend in the SYNOPSIS section.
572		Where additional semantics apply to a function, the material is identified by use of the TMO
573		margin legend.
574	TMR	Timers
575		The functionality described is optional. The functionality described is also an extension to the
576		ISO C standard.
577		Where applicable, functions are marked with the TMR margin legend in the SYNOPSIS section.
578		Where additional semantics apply to a function, the material is identified by use of the TMR
579		margin legend.
580	TPI	Threads Priority Inheritance
581		The functionality described is optional. The functionality described is also an extension to the
582		ISO C standard.
583		Where applicable, functions are marked with the TPI margin legend in the SYNOPSIS section.
584		Where additional semantics apply to a function, the material is identified by use of the TPI
585		margin legend.
586	TPP	Thread Priority Protection
587		The functionality described is optional. The functionality described is also an extension to the
588		ISO C standard.
589		Where applicable, functions are marked with the TPP margin legend in the SYNOPSIS section.
590		Where additional semantics apply to a function, the material is identified by use of the TPP
591		margin legend.
592	TPS	Thread Execution Scheduling
593		The functionality described is optional. The functionality described is also an extension to the
594		ISO C standard.
595		Where applicable, functions are marked with the TPS margin legend for the SYNOPSIS section.
596		Where additional semantics apply to a function, the material is identified by use of the TPS
597		margin legend.
598	TSA	Thread Stack Address Attribute
599		The functionality described is optional. The functionality described is also an extension to the
600		ISO C standard.
601		Where applicable, functions are marked with the TPS margin legend for the SYNOPSIS section.
602		Where additional semantics apply to a function, the material is identified by use of the TSA
603		margin legend.
604	TSF	Thread-Safe Functions
605		The functionality described is optional. The functionality described is also an extension to the
606		ISO C standard.
607		Where applicable, functions are marked with the TSF margin legend in the SYNOPSIS section.
608		Where additional semantics apply to a function, the material is identified by use of the TSF
609		margin legend.
610	TSH	Thread Process-Shared Synchronization
611		The functionality described is optional. The functionality described is also an extension to the
612		ISO C standard.

613		Where applicable, functions are marked with the TSH margin legend in the SYNOPSIS section.
614		Where additional semantics apply to a function, the material is identified by use of the TSH
615		margin legend.
616	TSP	Thread Sporadic Server
617		The functionality described is optional. The functionality described is also an extension to the
618		ISO C standard.
619		Where applicable, functions are marked with the TSP margin legend in the SYNOPSIS section.
620		Where additional semantics apply to a function, the material is identified by use of the TSP
621		margin legend.
622	TSS	Thread Stack Address Size
623		The functionality described is optional. The functionality described is also an extension to the
624		ISO C standard.
625		Where applicable, functions are marked with the TSS margin legend in the SYNOPSIS section.
626		Where additional semantics apply to a function, the material is identified by use of the TSS
627		margin legend.
628	TYM	Typed Memory Objects
629		The functionality described is optional. The functionality described is also an extension to the
630		ISO C standard.
631		Where applicable, functions are marked with the TYM margin legend in the SYNOPSIS section.
632		Where additional semantics apply to a function, the material is identified by use of the TYM
633		margin legend.
634	UN	Possibly Unsupportable Feature
635		The functionality described is an XSI extension. It need not be possible to implement the
636		required functionality (as defined) on all conformant systems and the functionality need not be
637		present. This may, for example, be the case where the conformant system is hosted and the
638		underlying system provides the service in an alternative way.
639	UP	User Portability Utilities
640		The functionality described is optional.
641		Where applicable, utilities are marked with the UP margin legend in the SYNOPSIS section.
642		Where additional semantics apply to a utility, the material is identified by use of the UP margin
643		legend.
644	XSI	Extension
645		The functionality described is an XSI extension. Functionality marked XSI is also an extension to
646		the ISO C standard. Application writers may confidently make use of an extension on all
647		systems supporting the X/Open System Interfaces Extension.
648		If an entire SYNOPSIS section is shaded and marked with one XSI, all the functionality described
649		in that reference page is an extension. See the System Interface Definitions volume of
650		IEEE Std. 1003.1-200x, Section 3.436, XSI.
651	XSR	XSI STREAMS
652		The functionality described is optional. The functionality described is also an extension to the
653		ISO C standard.
654		Where applicable, functions are marked with the XSR margin legend in the SYNOPSIS section.
655		Where additional semantics apply to a function, the material is identified by use of the XSR
656		margin legend.

1.9 Utility Limits

This section lists magnitude limitations imposed by a specific implementation. The braces notation, {LIMIT}, is used in this volume of IEEE Std. 1003.1-200x to indicate these values, but the braces are not part of the name.

Table 1-1 Utility Limit Minimum Values

Name	Description	Value
{POSIX2_BC_BASE_MAX}	The maximum <i>obase</i> value allowed by the <i>bc</i> utility.	99
{POSIX2_BC_DIM_MAX}	The maximum number of elements permitted in an array by the <i>bc</i> utility.	2048
{POSIX2_BC_SCALE_MAX}	The maximum <i>scale</i> value allowed by the <i>bc</i> utility.	99
{POSIX2_BC_STRING_MAX}	The maximum length of a string constant accepted by the <i>bc</i> utility.	1000
{POSIX2_COLL_WEIGHTS_MAX}	The maximum number of weights that can be assigned to an entry of the <i>LC_COLLATE</i> order keyword in the locale definition file; see the border_start keyword in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 7.3.2, <i>LC_COLLATE</i> .	2
{POSIX2_EXPR_NEST_MAX}	The maximum number of expressions that can be nested within parentheses by the <i>expr</i> utility.	32
{POSIX2_LINE_MAX}	Unless otherwise noted, the maximum length, in bytes, of the input line of a utility (either standard input or another file), when the utility is described as processing text files. The length includes room for the trailing newline.	2048
{POSIX2_RE_DUP_MAX}	The maximum number of repeated occurrences of a BRE permitted when using the interval notation $\{m,n\}$; see the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 9.3.6, BREs Matching Multiple Characters.	255
{POSIX2_VERSION}	This value indicates the version of the utilities in this volume of IEEE Std. 1003.1-200x that are provided by the implementation. It changes with each published version.	199209

The values specified in Table 1-1 represent the lowest values conforming implementations shall provide and, consequently, the largest values on which an application can rely without further enquiries, as described below. These values shall be accessible to applications via the *getconf* utility (see *getconf* on page 517) and through the *sysconf()* function defined in the System Interfaces volume of IEEE Std. 1003.1-200x. The literal names shown in Table 1-1 apply only to the *getconf* utility; the high-level language binding describes the exact form of each name to be used by the interfaces in that binding.

Implementations may provide more liberal, or less restrictive, values than shown in Table 1-1. These possibly more liberal values are accessible using the symbols in Table 1-2 on page 22.

The `sysconf()` function defined in the System Interfaces volume of IEEE Std. 1003.1-200x or the `getconf` utility return the value of each symbol on each specific implementation. The value so retrieved is the largest, or most liberal, value that is available throughout the session lifetime, as determined at session creation. The literal names shown in the table apply only to the `getconf` utility; the high-level language binding describes the exact form of each name to be used by the interfaces in that binding.

All numeric limits defined by the System Interfaces volume of IEEE Std. 1003.1-200x, such as `{PATH_MAX}`, also apply to this volume of IEEE Std. 1003.1-200x. All the utilities defined by this volume of IEEE Std. 1003.1-200x are implicitly limited by these values, unless otherwise noted in the utility descriptions.

It is not guaranteed that the application can in fact push a value to the specified limit of an implementation in any given case, or at all, as a lack of virtual memory or other resources may prevent this. The limit value indicates only that the implementation does not specifically impose any arbitrary, more restrictive limit.

Table 1-2 Symbolic Utility Limits

Name	Description	Minimum Value
<code>{BC_BASE_MAX}</code>	The maximum <i>obase</i> value allowed by the <i>bc</i> utility.	<code>{POSIX2_BC_BASE_MAX}</code>
<code>{BC_DIM_MAX}</code>	The maximum number of elements permitted in an array by the <i>bc</i> utility.	<code>{POSIX2_BC_DIM_MAX}</code>
<code>{BC_SCALE_MAX}</code>	The maximum <i>scale</i> value allowed by the <i>bc</i> utility.	<code>{POSIX2_BC_SCALE_MAX}</code>
<code>{BC_STRING_MAX}</code>	The maximum length of a string constant accepted by the <i>bc</i> utility.	<code>{POSIX2_BC_STRING_MAX}</code>
<code>{COLL_WEIGHTS_MAX}</code>	The maximum number of weights that can be assigned to an entry of the <i>LC_COLLATE</i> order keyword in the locale definition file; see the order_start keyword in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 7.3.2, <i>LC_COLLATE</i> .	<code>{POSIX2_COLL_WEIGHTS_MAX}</code>
<code>{EXPR_NEST_MAX}</code>	The maximum number of expressions that can be nested within parentheses by the <i>expr</i> utility.	<code>{POSIX2_EXPR_NEST_MAX}</code>
<code>{LINE_MAX}</code>	Unless otherwise noted, the maximum length, in bytes, of the input line of a utility (either standard input or another file), when the	<code>{POSIX2_LINE_MAX}</code>

Name	Description	Minimum Value
{RE_DUP_MAX}	utility is described as processing text files. The length includes room for the trailing newline. The maximum number of repeated occurrences of a BRE permitted when using the interval notation $\{m,n\}$; see the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 9.3.6, BREs Matching Multiple Characters.	{POSIX2_RE_DUP_MAX}

The following value may be a constant within an implementation or may vary from one path name to another.

{POSIX2_SYMLINKS}

When referring to a directory, the system supports the creation of symbolic links within that directory; for non-directory files, the meaning of {POSIX2_SYMLINKS} is undefined.

Rationale

The {POSIX2_SYMLINKS} variable indicates that the underlying operating system supports the creation of symbolic links in specific directories. Many of the utilities defined in the IEEE Std. 1003.1-200x that deal with symbolic links do not depend on this value. For example, a utility that follows symbolic links (or does not, as the case may be) will only be affected by a symbolic link if it encounters one. Presumably, a file system that does not support symbolic links will not contain any. This variable does affect such utilities as *ln -s* and *pax* that attempt to create symbolic links.

{POSIX2_SYMLINKS} was developed even though there is no comparable configuration value in the IEEE P1003.1a draft standard.

Notes to Reviewers

This section with side shading will not appear in the final copy. - Ed.

Do we want to add POSIX2_SYMLINKS to `<unistd.h>/sysconf()` and remove the last sentence?

1.10 Grammar Conventions

Portions of this volume of IEEE Std. 1003.1-200x are expressed in terms of a special grammar notation. It is used to portray the complex syntax of certain program input. The grammar is based on the syntax used by the *yacc* utility. However, it does not represent fully functional *yacc* input, suitable for program use; the lexical processing and all semantic requirements are described only in textual form. The grammar is not based on source used in any traditional implementation and has not been tested with the semantic code that would normally be required to accompany it. Furthermore, there is no implication that the partial *yacc* code presented represents the most efficient, or only, means of supporting the complex syntax within the utility. Implementations may use other programming languages or algorithms, as long as the syntax supported is the same as that represented by the grammar.

The following typographical conventions are used in the grammar; they have no significance except to aid in reading.

- The identifiers for the reserved words of the language are shown with a leading capital letter. (These are terminals in the grammar; for example, **While**, **Case**.)
- The identifiers for terminals in the grammar are all named with uppercase letters and underscores; for example, **NEWLINE**, **ASSIGN_OP**, **NAME**.
- The identifiers for non-terminals are all lowercase.

1.11 Utility Description Defaults

This section describes all of the subsections used within the utility descriptions, including:

- Intended usage of the section
- Global defaults that affect all the standard utilities
- The meanings of notations used in this volume of IEEE Std. 1003.1-200x that are specific to individual utility sections

Integer variables and constants, including the values of operands and option-arguments, used by the utilities listed in this volume of IEEE Std. 1003.1-200x shall be implemented as equivalent to the ISO C standard **signed long** data type. Conversion between types shall be as described in the ISO C standard. The evaluation of arithmetic expressions shall be equivalent to that described in Section 6.3 of the ISO C standard.

NAME

This section gives the name or names of the utility and briefly states its purpose.

SYNOPSIS

The SYNOPSIS section summarizes the syntax of the calling sequence for the utility, including options, option-arguments, and operands. Standards for utility naming are described in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines; for describing the utility's arguments in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 12.1, Utility Argument Syntax.

DESCRIPTION

The DESCRIPTION section describes the actions of the utility. If the utility has a very complex set of subcommands or its own procedural language, an EXTENDED DESCRIPTION section is also provided. Most explanations of optional functionality are omitted here, as they are usually explained in the OPTIONS section.

Some utilities in this volume of IEEE Std. 1003.1-200x are described in terms of functionality equivalent to the System Interfaces volume of IEEE Std. 1003.1-200x. When specific functions are cited, the underlying operating system provides equivalent functionality and all side effects associated with successful execution of the function. The treatment of errors and intermediate results from the individual functions cited is generally not specified by this volume of IEEE Std. 1003.1-200x. See the utility's EXIT STATUS and CONSEQUENCES OF ERRORS sections for all actions associated with errors encountered by the utility.

OPTIONS

The OPTIONS section describes the utility options and option-arguments, and how they modify the actions of the utility. Standard utilities that have options either fully comply with the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines or describe all deviations. Apparent disagreements between functionality descriptions in the OPTIONS and DESCRIPTION (or EXTENDED DESCRIPTION) sections are always resolved in favor of the OPTIONS section.

Each OPTIONS section that uses the phrase "The ... utility supports the Utility Syntax Guidelines ..." refers only to the use of the utility as specified by this volume of IEEE Std. 1003.1-200x; implementation extensions should also conform to the guidelines, but may allow exceptions for historical practice.

Unless otherwise stated in the utility description, when given an option unrecognized by the implementation, or when a required option-argument is not provided, standard utilities issue a diagnostic message to standard error and exit with a non-zero exit status.

XSI All utilities in this volume of IEEE Std. 1003.1-200x are capable of processing arguments using 8-bit transparency.

Default Behavior: When this section is listed as “None.”, it means that the implementation need not support any options. Standard utilities that do not accept options, but that do accept operands, recognize “—” as a first argument to be discarded.

The requirement for recognizing “—” is because portable applications need a way to shield their operands from any arbitrary options that the implementation may provide as an extension. For example, if the standard utility *foo* is listed as taking no options, and the application needed to give it a path name with a leading hyphen, it could safely do it as:

```
foo — -myfile
```

and avoid any problems with *-m* used as an extension.

OPERANDS

The OPERANDS section describes the utility operands, and how they affect the actions of the utility. Apparent disagreements between functionality descriptions in the OPERANDS and DESCRIPTION (or EXTENDED DESCRIPTION) sections are always resolved in favor of the OPERANDS section.

If an operand naming a file can be specified as ‘-’, which means to use the standard input instead of a named file, this is explicitly stated in this section. Unless otherwise stated, the use of multiple instances of ‘-’ to mean standard input in a single command produces unspecified results.

Unless otherwise stated, the standard utilities that accept operands process those operands in the order specified in the command line.

Default Behavior: When this section is listed as “None.”, it means that the implementation need not support any operands.

STDIN

The STDIN section describes the standard input of the utility. This section is frequently merely a reference to the following section, as many utilities treat standard input and input files in the same manner. Unless otherwise stated, all restrictions described in the INPUT FILES section apply to this section as well.

Use of a terminal for standard input can cause any of the standard utilities that read standard input to stop when used in the background. For this reason, applications should not use interactive features in scripts to be placed in the background.

The specified standard input format of the standard utilities does not depend on the existence or value of the environment variables defined in this volume of IEEE Std. 1003.1-200x, except as provided by this volume of IEEE Std. 1003.1-200x.

Default Behavior: When this section is listed as “Not used.”, it means that the standard input is not read when the utility is used as described by this volume of IEEE Std. 1003.1-200x.

INPUT FILES

The INPUT FILES section describes the files, other than the standard input, used as input by the utility. It includes files named as operands and option-arguments as well as other files that are referred to, such as start-up and initialization files, databases, and so on. Commonly-used files are generally described in one place and cross-referenced by other utilities.

XSI All utilities in this volume of IEEE Std. 1003.1-200x are capable of processing input files using 8-bit transparency.

When a standard utility reads a seekable input file and terminates without an error before it reaches end-of-file, the utility ensures that the file offset in the open file description is properly positioned just past the last byte processed by the utility. For files that are not seekable, the state of the file offset in the open file description for that file is unspecified. A portable application cannot assume that the following three commands are equivalent:

```
tail -n +2 file
(sed -n 1q; cat) < file
cat file | (sed -n 1q; cat)
```

The second command is equivalent to the first only when the file is seekable. The third command leaves the file offset in the open file description in an unspecified state. Other utilities, such as *head*, *read*, and *sh*, have similar properties.

Some of the standard utilities, such as filters, process input files a line or a block at a time and have no restrictions on the maximum input file size. Some utilities may have size limitations that are not as obvious as file space or memory limitations. Such limitations should reflect resource limitations of some sort, not arbitrary limits set by implementors. Implementations document those utilities that are limited by constraints other than file system space, available memory, and other limits specifically cited by this volume of IEEE Std. 1003.1-200x, and identify what the constraint is and indicate a way of estimating when the constraint would be reached. Similarly, some utilities descend the directory tree (recursively). Implementations also document any limits that they may have in descending the directory tree that are beyond limits cited by this volume of IEEE Std. 1003.1-200x.

When an input file is described as a *text file*, the utility produces undefined results if given input that is not from a text file, unless otherwise stated. Some utilities (for example, *make*, *read*, *sh*) allow for continued input lines using an escaped <newline> convention; unless otherwise stated, the utility need not be able to accumulate more than {LINE_MAX} bytes from a set of multiple, continued input lines. Thus, for a portable application the total of all the continued lines in a set cannot exceed {LINE_MAX}. If a utility using the escaped <newline> convention detects an end-of-file condition immediately after an escaped <newline>, the results are unspecified.

Record formats are described in a notation similar to that used by the C-language function, *printf()*. See the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 5, File Format Notation for a description of this notation. The format description is intended to be sufficiently rigorous to allow other applications to generate these input files. However, since <blank> characters can legitimately be included in some of the fields described by the standard utilities, particularly in locales other than the POSIX locale, this intent is not always realized.

Default Behavior: When this section is listed as “None.”, it means that no input files are required to be supplied when the utility is used as described by this volume of

IEEE Std. 1003.1-200x.

ENVIRONMENT VARIABLES

The ENVIRONMENT VARIABLES section lists what variables affect the utility's execution.

The entire manner in which environment variables described in this volume of IEEE Std. 1003.1-200x affect the behavior of each utility is described in the ENVIRONMENT VARIABLES section for that utility, in conjunction with the global effects of the *LANG*, *LC_ALL*, and *NLSPATH* environment variables described in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 8, Environment Variables. The existence or value of environment variables described in this volume of IEEE Std. 1003.1-200x do not otherwise affect the specified behavior of the standard utilities. Any effects of the existence or value of environment variables not described by this volume of IEEE Std. 1003.1-200x upon the standard utilities are unspecified.

For those standard utilities that use environment variables as a means for selecting a utility to execute (such as *CC* in *make*), the string provided to the utility is subjected to the path search described for *PATH* in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 8, Environment Variables.

All utilities in this volume of IEEE Std. 1003.1-200x are capable of processing environment variable names and values using 8-bit transparency.

Default Behavior: When this section is listed as “None.”, it means that the behavior of the utility is not directly affected by environment variables described by this volume of IEEE Std. 1003.1-200x when the utility is used as described by this volume of IEEE Std. 1003.1-200x.

ASYNCHRONOUS EVENTS

The ASYNCHRONOUS EVENTS section lists how the utility reacts to such events as signals and what signals are caught.

Default Behavior: When this section is listed as “Default.”, or it refers to “the standard action for all other signals; see Section 1.11 on page 25” it means that the action taken as a result of the signal is one of the following:

1. The action is that inherited from the parent according to the rules of inheritance of signal actions defined in the System Interfaces volume of IEEE Std. 1003.1-200x.
2. When no action has been taken to change the default, the default action is that specified by the System Interfaces volume of IEEE Std. 1003.1-200x.
3. The result of the utility's execution is as if default actions had been taken.

A utility is permitted to catch a signal, perform some additional processing (such as deleting temporary files), restore the default signal action (or action inherited from the parent process), and resignal itself.

STDOUT

The STDOUT section describes the standard output of the utility. This section is frequently merely a reference to the following section, OUTPUT FILES, because many utilities treat standard output and output files in the same manner.

Use of a terminal for standard output may cause any of the standard utilities that write standard output to stop when used in the background. For this reason, applications should not use interactive features in scripts to be placed in the background.

Record formats are described in a notation similar to that used by the C-language function, *printf()*. See the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 5, File Format Notation for a description of this notation.

The specified standard output of the standard utilities does not depend on the existence or value of the environment variables defined in this volume of IEEE Std. 1003.1-200x, except as provided by this volume of IEEE Std. 1003.1-200x.

Some of the standard utilities describe their output using the verb *display*, defined in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 3.135, Display. Output described in the STDOUT sections of such utilities may be produced using means other than standard output. When standard output is directed to a terminal, the output described is written directly to the terminal. Otherwise, the results are undefined.

Default Behavior: When this section is listed as “Not used.”, it means that the standard output is not written when the utility is used as described by this volume of IEEE Std. 1003.1-200x.

1000 STDERR

The STDERR section describes the standard error output of the utility. Only those messages that are purposely sent by the utility are described.

Use of a terminal for standard error may cause any of the standard utilities that write standard error output to stop when used in the background. For this reason, applications should not use interactive features in scripts to be placed in the background.

The format of diagnostic messages for most utilities is unspecified, but the language and cultural conventions of diagnostic and informative messages whose format is unspecified by this volume of IEEE Std. 1003.1-200x should be affected by the setting of *LC_MESSAGES* and *NLSPATH*.

The specified standard error output of standard utilities does not depend on the existence or value of the environment variables defined in this volume of IEEE Std. 1003.1-200x, except as provided by this volume of IEEE Std. 1003.1-200x.

Default Behavior: When this section is listed as “Used only for diagnostic messages.”, it means that, unless otherwise stated, the diagnostic messages are sent to the standard error only when the exit status is non-zero and the utility is used as described by this volume of IEEE Std. 1003.1-200x.

When this section is listed as “Not used.”, it means that the standard error is not used when the utility is used as described in this volume of IEEE Std. 1003.1-200x.

This section does not describe error messages that refer to incorrect operation of the utility. Consider a utility that processes program source code as its input. This section is used to describe messages produced by a correctly operating utility that encounters an error in the program source code on which it is processing. However, a message indicating that the utility had insufficient memory in which to operate would not be described.

Some utilities have traditionally produced warning messages without returning a non-zero exit status; these are specifically noted in their sections. Other utilities are expected to remain absolutely quiet on the standard error if they want to return zero, unless the implementation provides some sort of extension to increase the verbosity or debugging level.

OUTPUT FILES

The OUTPUT FILES section describes the files created or modified by the utility. Temporary or system files that are created for internal usage by this utility or other parts of the implementation (for example, spool, log, and audit files) are not described in this, or any, section. The utilities creating such files and the names of such files are unspecified. If applications are written to use temporary or intermediate files, they should use the *TMPDIR* environment variable, if it is set and represents an accessible directory, to select the location of temporary files.

Temporary files used by the standard utilities are named so that different utilities or multiple instances of the same utility can operate simultaneously without regard to their working directories, or any other process characteristic other than process ID. There are two exceptions to this rule:

1. Resources for temporary files other than the name space (for example, disk space, available directory entries, or number of processes allowed) are not guaranteed.
2. Certain standard utilities generate output files that are intended as input for other utilities (for example, *lex* generates *lex.yy.c*), and these cannot have unique names. These cases are explicitly identified in the descriptions of the respective utilities.

Any temporary file created by the implementation is removed by the implementation upon a utility's successful exit, exit because of errors, or before termination by any of the SIGHUP, SIGINT, or SIGTERM signals, unless specified otherwise by the utility description.

Receipt of the SIGQUIT signal should generally cause termination (unless in some debugging mode) that would bypass any attempted recovery actions.

Record formats are described in a notation similar to that used by the C-language function, *printf()*; see the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 5, File Format Notation for a description of this notation.

Default Behavior: When this section is listed as "None.", it means that no files are created or modified as a consequence of direct action on the part of the utility when the utility is used as described by this volume of IEEE Std. 1003.1-200x. However, the utility may create or modify system files, such as log files, that are outside the utility's normal execution environment.

EXTENDED DESCRIPTION

The EXTENDED DESCRIPTION section provides a place for describing the actions of very complicated utilities, such as text editors or language processors, which typically have elaborate command languages.

Default Behavior: When this section is listed as "None.", no further description is necessary.

EXIT STATUS

The EXIT STATUS section describes the values the utility returns to the calling program, or shell, and the conditions that cause these values to be returned. Usually, utilities return zero for successful completion and values greater than zero for various error conditions. If specific numeric values are listed in this section, the system uses those values for the errors described. In some cases, status values are listed more loosely, such as *>0*. A portable application cannot rely on any specific value in the range shown and shall be prepared to receive any value in the range.

For example, a utility may list zero as a successful return, 1 as a failure for a specific reason, and >1 as “an error occurred”. In this case, unspecified conditions may cause a 2 or 3, or other value, to be returned. A portable application should be written so that it tests for successful exit status values (zero in this case), rather than relying upon the single specific error value listed in this volume of IEEE Std. 1003.1-200x. In that way, it has maximum portability, even on implementations with extensions.

Unspecified error conditions may be represented by specific values not listed in this volume of IEEE Std. 1003.1-200x.

CONSEQUENCES OF ERRORS

The CONSEQUENCES OF ERRORS section describes the effects on the environment, file systems, process state, and so on, when error conditions occur. It does not describe error messages produced or exit status values used.

The many reasons for failure of a utility are generally not specified by the utility descriptions. Utilities may terminate prematurely if they encounter: invalid usage of options, arguments, or environment variables; invalid usage of the complex syntaxes expressed in EXTENDED DESCRIPTION sections; difficulties accessing, creating, reading, or writing files; or difficulties associated with the privileges of the process.

The following apply to each utility, unless otherwise stated:

- If the requested action cannot be performed on an operand representing a file, directory, user, process, and so on, the utility issues a diagnostic message to standard error and continues processing the next operand in sequence, but the final exit status is returned as non-zero.

For a utility that recursively traverses a file hierarchy (such as *find* or *chown -R*), if the requested action cannot be performed on a file or directory encountered in the hierarchy, the utility issues a diagnostic message to standard error and continues processing the remaining files in the hierarchy, but the final exit status is returned as non-zero.

- If the requested action characterized by an option or option-argument cannot be performed, the utility issues a diagnostic message to standard error and the exit status returned is non-zero.
- When an unrecoverable error condition is encountered, the utility exits with a non-zero exit status.
- A diagnostic message is written to standard error whenever an error condition occurs.

When a utility encounters an error condition several actions are possible, depending on the severity of the error and the state of the utility. Included in the possible actions of various utilities are: deletion of temporary or intermediate work files; deletion of incomplete files; validity checking of the file system or directory.

Default Behavior: When this section is listed as “Default.”, it means that any changes to the environment are unspecified.

APPLICATION USAGE

This section is non-normative.

The APPLICATION USAGE section gives advice to the application programmer or user about the way the utility should be used.

EXAMPLES

This section is non-normative.

The EXAMPLES section gives one or more examples of usage, where appropriate. In the event of conflict between an example and a normative part of the specification, the normative material is to be taken as correct.

In all examples, quoting has been used, showing how sample commands (utility names combined with arguments) could be passed correctly to a shell (see *sh*) or as a string to the *system()* function defined in the System Interfaces volume of IEEE Std. 1003.1-200x. Such quoting would not be used if the utility is invoked using one of the *exec* functions defined in the System Interfaces volume of IEEE Std. 1003.1-200x.

RATIONALE

This section is non-normative.

This section contains historical information concerning the contents of this volume of IEEE Std. 1003.1-200x and why features were included or discarded by the standard developers.

FUTURE DIRECTIONS

This section is non-normative.

The FUTURE DIRECTIONS section should be used as a guide to current thinking; there is not necessarily a commitment to implement all of these future directions in their entirety.

SEE ALSO

This section is non-normative.

The SEE ALSO section lists related entries.

CHANGE HISTORY

This section is non-normative.

The CHANGE HISTORY section shows the derivation of the description used by this volume of IEEE Std. 1003.1-200x and lists the functional differences between Issues 4 and 6.

Certain of the standard utilities describe how they can invoke other utilities or applications, such as by passing a command string to the command interpreter. The external influences (STDIN, ENVIRONMENT VARIABLES, and so on) and external effects (STDOUT, CONSEQUENCES OF ERRORS, and so on) of such invoked utilities are not described in the section concerning the standard utility that invokes them.

1.12 Considerations for Utilities in Support of Files of Arbitrary Size

The following utilities support files of any size up to the maximum that can be created by the implementation. This support includes correct writing of file size-related values (such as file sizes and offsets, line numbers, and block counts) and correct interpretation of command line arguments that contain such values.

<i>basename</i>	Return non-directory portion of path name.
<i>cat</i>	Concatenate and print files.
<i>cd</i>	Change working directory.
<i>chgrp</i>	Change file group ownership.
<i>chmod</i>	Change file modes.
<i>chown</i>	Change file ownership.
<i>cksum</i>	Write file checksums and sizes.
<i>cmp</i>	Compare two files.
<i>cp</i>	Copy files.
<i>dd</i>	Convert and copy a file.
<i>df</i>	Report free disk space.
<i>dirname</i>	Return directory portion of path name.
<i>du</i>	Estimate file space usage.
<i>find</i>	Find files.
<i>ln</i>	Link files.
<i>ls</i>	List directory contents.
<i>mkdir</i>	Make directories.
<i>mv</i>	Move files.
<i>pathchk</i>	Check path names.
<i>pwd</i>	Return working directory name.
<i>rm</i>	Remove directory entries.
<i>rmdir</i>	Remove directories.
<i>sh</i>	Shell, the standard command language interpreter.
<i>sum</i>	Print checksum and block or byte count of a file.
<i>test</i>	Evaluate expression.
<i>touch</i>	Change file access and modification times.
<i>ulimit</i>	Set or report file size limit.

Exceptions to the requirement that utilities support files of any size up to the maximum are:

1. Uses of files as command scripts, or for configuration or control, are exempt. For example, it is not required that *sh* be able to read an arbitrarily large **.profile**.
2. Shell input and output redirection are exempt. For example, it is not required that the redirections *sum < file* or *echo foo > file* succeed for an arbitrarily large existing file.

Shell Command Language

This chapter contains the definition of the Shell Command Language.

2.1 Shell Introduction

The shell is a command language interpreter. This chapter describes the syntax of that command language as it is used by the *sh* utility and the *system()* and *popen()* functions defined in the System Interfaces volume of IEEE Std. 1003.1-200x.

The shell operates according to the following general overview of operations. The specific details are included in the cited sections of this chapter.

1. The shell reads its input from a file (see *sh*), from the *-c* option or from the *system()* and *popen()* functions defined in the System Interfaces volume of IEEE Std. 1003.1-200x. If the first line of a file of shell commands starts with the characters "#!", the results are unspecified.
2. The shell breaks the input into tokens: words and operators; see Section 2.3 on page 39.
3. The shell parses the input into simple commands (see Section 2.9.1 on page 67) and compound commands (see Section 2.9.4 on page 75).
4. The shell performs various expansions (separately) on different parts of each command, resulting in a list of path names and fields to be treated as a command and arguments; see Section 2.6 on page 49.
5. The shell performs redirection (see Section 2.7 on page 60) and removes redirection operators and their operands from the parameter list.
6. The shell executes a function (see Section 2.9.5 on page 79), built-in (see Section 2.14 on page 96), executable file, or script, giving the names of the arguments as positional parameters numbered 1 to *n*, and the name of the command (or in the case of a function within a script, the name of the script) as the positional parameter numbered 0 (see Section 2.9.1.1 on page 69).
7. The shell optionally waits for the command to complete and collects the exit status (see Section 2.8.2 on page 65).

Rationale

The System V shell was selected as the starting point for this volume of IEEE Std. 1003.1-200x. The BSD C shell was excluded from consideration for the following reasons:

- Most historically portable shell scripts assume the Version 7 Bourne shell, from which the System V shell is derived.
- The majority of tutorial materials on shell programming assume the System V shell.

The construct "#!" is reserved for implementations wishing to provide that extension. If it were not reserved, this volume of IEEE Std. 1003.1-200x would disallow it by forcing it to be a comment. As it stands, a conforming application must not use "#!" as the first two characters of the file.

2.2 Quoting

Quoting is used to remove the special meaning of certain characters or words to the shell. Quoting can be used to preserve the literal meaning of the special characters in the next paragraph, prevent reserved words from being recognized as such, and prevent parameter expansion and command substitution within here-document processing (see Section 2.7.4 on page 61).

The application shall quote the following characters if they are to represent themselves:

| & ; < > () \$ ' \ " ' <space> <tab> <newline>

and the following may need to be quoted under certain circumstances. That is, these characters may be special depending on conditions described elsewhere in this volume of IEEE Std. 1003.1-200x:

* ? [# ~ = %

The various quoting mechanisms are the escape character, single-quotes, and double-quotes. The here-document represents another form of quoting; see Section 2.7.4 on page 61.

2.2.1 Escape Character (Backslash)

A backslash that is not quoted shall preserve the literal value of the following character, with the exception of a <newline> character. If a <newline> character follows the backslash, the shell shall interpret this as line continuation. The backslash and <newline> characters shall be removed before splitting the input into tokens. Since the escaped <newline> character is removed entirely from the input and is not replaced by any white space, it cannot serve as a token separator.

2.2.2 Single-Quotes

Enclosing characters in single-quotes (' ') shall preserve the literal value of each character within the single-quotes. A single-quote cannot occur within single-quotes.

Rationale

A backslash cannot be used to escape a single-quote in a single-quoted string. An embedded quote can be created by writing, for example: " 'a' \ ' 'b' ", which yields "a'b". (See Section 2.6.5 on page 58 for a better understanding of how portions of words are either split into fields or remain concatenated.) A single token can be made up of concatenated partial strings containing all three kinds of quoting or escaping, thus permitting any combination of characters.

2.2.3 Double-Quotes

Enclosing characters in double-quotes (" ") shall preserve the literal value of all characters within the double-quotes, with the exception of the characters dollar sign, backquote, and backslash, as follows:

\$ The dollar sign shall retain its special meaning introducing parameter expansion (see Section 2.6.2 on page 51), a form of command substitution (see Section 2.6.3 on page 54), and arithmetic expansion (see Section 2.6.4 on page 56).

The input characters within the quoted string that are also enclosed between "\$ (" and the matching ') ' is not affected by the double-quotes, but rather shall define that command whose output replaces the "\$ (. . .)" when the word is expanded. The tokenizing rules in Section 2.3 on page 39 shall be applied recursively to find the matching ') '.

1270 Within the string of characters from an enclosed "\${" to the matching "}", an even number
 1271 of unescaped double-quotes or single-quotes, if any, shall occur. A preceding backslash
 1272 character shall be used to escape a literal '{' or '}'. The rule in Section 2.6.2 on page 51
 1273 shall be used to determine the matching '}'.

1274 ` The backquote shall retain its special meaning introducing the other form of command
 1275 substitution (see Section 2.6.3 on page 54). The portion of the quoted string from the initial
 1276 backquote and the characters up to the next backquote that is not preceded by a backslash,
 1277 having escape characters removed, defines that command whose output replaces "`..."
 1278 when the word is expanded. Either of the following cases produces undefined results:

- 1279 • A single-quoted or double-quoted string that begins, but does not end, within the
 1280 "`..." sequence

- 1281 • A "`..." sequence that begins, but does not end, within the same double-quoted
 1282 string

1283 \ The backslash shall retain its special meaning as an escape character (see Section 2.2.1 on
 1284 page 36) only when followed by one of the following characters when considered special:

1285 \$ ` " \ <newline>

1286 The application shall ensure that a double-quote is preceded by a backslash to be included
 1287 within double-quotes. The parameter '@' has special meaning inside double-quotes and is
 1288 described in Section 2.5.2 on page 43.

1289 Rationale

1290 The escaped <newline> used for line continuation is removed entirely from the input and is not
 1291 replaced by any white space. Therefore, it cannot serve as a token separator.

1292 In double-quoting, if a backslash is immediately followed by a character that would be
 1293 interpreted as having a special meaning, the backslash is deleted and the subsequent character is
 1294 taken literally. If a backslash does not precede a character that would have a special meaning, it
 1295 is left in place unmodified and the character immediately following it is also left unmodified.
 1296 Thus, for example:

1297 "\\$" → \$

1298 "\a" → \a

1299 It would be desirable to include the statement “The characters from an enclosed "\${" to the
 1300 matching '}' shall not be affected by the double quotes”, similar to the one for "\$()".
 1301 However, historical practice in the System V shell prevents this.

1302 The requirement that double-quotes be matched inside "\${...}" within double-quotes and the
 1303 rule for finding the matching '}' in Section 2.6.2 on page 51 eliminate several subtle
 1304 inconsistencies in expansion for historical shells in rare cases; for example:

1305 "\${foo-bar}"

1306 yields **bar** when **foo** is not defined, and is an invalid substitution when **foo** is defined, in many
 1307 historical shells. The differences in processing the "\${...}" form have led to inconsistencies
 1308 between historical systems. A consequence of this rule is that single-quotes cannot be used to
 1309 quote the '}' within "\${...}"; for example:

1310 unset bar

1311 foo="\${bar-'}'"

is invalid because the "\${...}" substitution contains an unpaired unescaped single-quote. The backslash can be used to escape the '}' in this example to achieve the desired result:

```
unset bar
foo="${bar-\} }
```

The differences in processing the "\${...}" form have led to inconsistencies between the historical System V shell, BSD, and KornShells, and the text in this volume of IEEE Std. 1003.1-200x is an attempt to converge them without breaking too many applications. The only alternative to this compromise between shells would be to make the behavior unspecified whenever the literal characters ' ', '{', '}', and '"' appear within "\${...}". To write a portable script that uses these values, a user would have to assign variables, for example:

```
squote=\' dquote=\" lbrace='{ rbrace=}'
${foo-$squote$rbrace$squote}
```

rather than:

```
${foo-" ' ' ' " }
```

Some systems have allowed the end of the word to terminate the backquoted command substitution, such as in:

```
"`echo hello`"
```

This usage is undefined; the matching backquote is required by this volume of IEEE Std. 1003.1-200x. The other undefined usage can be illustrated by the example:

```
sh -c '` echo "foo`'
```

The description of the recursive actions involving command substitution can be illustrated with an example. Upon recognizing the introduction of command substitution, the shell parses input (in a new context), gathering the source for the command substitution until an unbalanced ')' or '' is located. For example, in the following:

```
echo "$(date; echo "
one" )"
```

the double-quote following the *echo* does not terminate the first double-quote; it is part of the command substitution script. Similarly, in:

```
echo "$(echo *)"
```

the asterisk is not quoted since it is inside command substitution; however:

```
echo "$(echo "*)"
```

is quoted (and represents the asterisk character itself).

2.3 Token Recognition

The shell reads its input in terms of lines from a file, from a terminal in the case of an interactive shell, or from a string in the case of *sh -c* or *system()*. The input lines can be of unlimited length. These lines are parsed using two major modes: ordinary token recognition and processing of here-documents.

When an **io_here** token has been recognized by the grammar (see Section 2.10 on page 82), one or more of the subsequent lines immediately following the next **NEWLINE** token form the body of one or more here-documents and shall be parsed according to the rules of Section 2.7.4 on page 61.

When it is not processing an **io_here**, the shell shall break its input into tokens by applying the first applicable rule below to the next character in its input. The token shall be from the current position in the input until a token is delimited according to one of the rules below; the characters forming the token are exactly those in the input, including any quoting characters. If it is indicated that a token is delimited, and no characters have been included in a token, processing shall continue until an actual token is delimited.

1. If the end of input is recognized, the current token shall be delimited. If there is no current token, the end-of-input indicator shall be returned as the token.
2. If the previous character was used as part of an operator and the current character is not quoted and can be used with the current characters to form an operator, it shall be used as part of that (operator) token.

Note that certain combinations of characters are invalid in portable scripts, as shown in the grammar, and that some systems have assigned these combinations (such as "`|&`") as valid control operators. Portable scripts cannot rely on receiving errors in all cases where this volume of IEEE Std. 1003.1-200x indicates that a syntax is invalid.

3. If the previous character was used as part of an operator and the current character cannot be used with the current characters to form an operator, the operator containing the previous character shall be delimited.
4. If the current character is backslash, single-quote, or double-quote (`'\'`, `'\''`, or `'\"'`) and it is not quoted, it shall affect quoting for subsequent characters up to the end of the quoted text. The rules for quoting are as described in Section 2.2 on page 36. During token recognition no substitutions shall be actually performed, and the result token shall contain exactly the characters that appear in the input (except for `<newline>` character joining), unmodified, including any embedded or enclosing quotes or substitution operators, between the quote mark and the end of the quoted text. The token shall not be delimited by the end of the quoted field.
5. If the current character is an unquoted `'$'` or `'\"'`, the shell shall identify the start of any candidates for parameter expansion (Section 2.6.2 on page 51), command substitution (Section 2.6.3 on page 54), or arithmetic expansion (Section 2.6.4 on page 56) from their introductory unquoted character sequences: `'$'` or `"${"`, `"$(` or `'\"'`, and `"$((`, respectively. The shell shall read sufficient input to determine the end of the unit to be expanded (as explained in the cited sections). While processing the characters, if instances of expansions or quoting are found nested within the substitution, the shell shall recursively process them in the manner specified for the construct that is found. The characters found from the beginning of the substitution to its end, allowing for any recursion necessary to recognize embedded constructs, shall be included unmodified in the result token, including any embedded or enclosing substitution operators or quotes. The token shall not be delimited by the end of the substitution.

6. If the current character is not quoted and can be used as the first character of a new operator, the current token (if any) shall be delimited. The current character shall be used as the beginning of the next (operator) token.
7. If the current character is an unquoted <newline> character, the current token shall be delimited.
8. If the current character is an unquoted <blank> character, any token containing the previous character is delimited and the current character shall be discarded.
9. If the previous character was part of a word, the current character shall be appended to that word.
10. If the current character is a '#', it and all subsequent characters up to, but excluding, the next <newline> character shall be discarded as a comment. The <newline> character that ends the line is not considered part of the comment. The '#' starts a comment only when it is at the beginning of a token. Since the search for the end-of-comment does not consider an escaped <newline> character specially, a comment cannot be continued to the next line.
11. The current character is used as the start of a new word.

Once a token is delimited, it is categorized as required by the grammar in Section 2.10 on page 82.

Rationale

The (3) rule about combining characters to form operators is not meant to preclude systems from extending the shell language when characters are combined in otherwise invalid ways. Portable applications cannot use invalid combinations, and test suites should not penalize systems that take advantage of this fact. For example, the unquoted combination "&" is not valid in a POSIX script, but has a specific KornShell meaning.

The (10) rule about '#' as the current character is the first in the sequence in which a new token is being assembled. The '#' starts a comment only when it is at the beginning of a token. This rule is also written to indicate that the search for the end-of-comment does not consider escaped <newline> specially, so that a comment cannot be continued to the next line.

2.3.1 Alias Substitution

UP XSI The processing of aliases shall be supported on all XSI-conformant systems or if the system supports the User Portability Utilities option.

After a token has been delimited, but before applying the grammatical rules in Section 2.10 on page 82, a resulting word that is identified to be the command name word of a simple command shall be examined to determine whether it is an unquoted, valid alias name. However, reserved words in correct grammatical context shall not be candidates for alias substitution. A valid alias name (see the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 3.10, Alias Name) shall be one that has been defined by the *alias* utility and not subsequently undefined using *unalias*. Implementations also may provide predefined valid aliases that are in effect when the shell is invoked. To prevent infinite loops in recursive aliasing, if the shell is not currently processing an alias of the same name, the word shall be replaced by the value of the alias; otherwise, it shall not be replaced.

If the value of the alias replacing the word ends in a <blank> character, the shell shall check the next command word for alias substitution; this process shall continue until a word is found that is not a valid alias or an alias value does not end in a <blank> character.

1435 When used as specified by this volume of IEEE Std. 1003.1-200x, alias definitions shall not be
1436 inherited by separate invocations of the shell or by the utility execution environments invoked
1437 by the shell; see Section 2.12 on page 90.

1438 **Rationale**

1439 The alias capability was added in the UPE because it is widely used in historical
1440 implementations by interactive users. It was omitted from the base standard because it is not
1441 commonly used in application scripts, particularly since aliases are not passed to child shells.

1442 The definition of *alias name* precludes an alias name containing a slash character. Since the text
1443 applies to the command words of simple commands, reserved words (in their proper places)
1444 cannot be confused with aliases.

1445 The placement of alias substitution in Token Recognition makes it clear that it precedes all of the
1446 word expansion steps.

1447 An example concerning trailing <blank> characters and reserved words follows. If the user
1448 types:

```
1449     $ alias foo="/bin/ls "  
1450     $ alias while="/"
```

1451 The effect of executing:

```
1452     $ while true  
1453     > do  
1454     > echo "Hello, World"  
1455     > done
```

1456 is a never-ending sequence of "Hello, World" strings to the screen. However, if the user
1457 types:

```
1458     $ foo while
```

1459 the result is an *ls* listing of /. Since the alias substitution for **foo** ends in a <space> character, the
1460 next word is checked for alias substitution. The next word, **while**, has also been aliased, so it is
1461 substituted as well. Since it is not in the proper position as a command word, it is not recognized
1462 as a reserved word.

1463 If the user types:

```
1464     $ foo; while
```

1465 **while** retains its normal reserved-word properties.

2.4 Reserved Words

Reserved words are words that have special meaning to the shell; see Section 2.9 on page 67. The following words shall be recognized as reserved words:

!	do	esac	in
{	done	fi	then
}	elif	for	until
case	else	if	while

This recognition shall only occur when none of the characters is quoted and when the word is used as:

- The first word of a command
- The first word following one of the reserved words other than **case**, **for**, or **in**
- The third word in a **case** or **for** command (only **in** is valid in this case)

See the grammar in Section 2.10 on page 82.

The following words may be recognized as reserved words on some systems (when none of the characters are quoted), causing unspecified results:

[[]]	function	select
-----------	-----------	-----------------	---------------

Words that are the concatenation of a name and a colon (':') are reserved; their use produces unspecified results. This reservation is to allow future implementations that support named labels for flow control.

Rationale

All reserved words are recognized syntactically as such in the contexts described. However, note that **in** is the only meaningful reserved word after a **case** or **for**; similarly, **in** is not meaningful as the first word of a simple command.

Reserved words are recognized only when they are delimited (that is, meet the definition of the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 3.432, Word), whereas operators are themselves delimiters. For instance, '(' and ')' are control operators, so that no <space> character is needed in (*list*). However, '{' and '}' are reserved words in {*list*}, so that in this case the leading <space> character and semicolon are required.

The list of unspecified reserved words is from the KornShell, so portable applications cannot use them in places a reserved word would be recognized. This list contained **time** in early proposals, but it was removed when the *time* utility was selected for this volume of IEEE Std. 1003.1-200x.

There was a strong argument for promoting braces to operators (instead of reserved words), so they would be syntactically equivalent to subshell operators. Concerns about compatibility outweighed the advantages of this approach. Nevertheless, portable applications should consider quoting '{' and '}' when they represent themselves.

The restriction on ending a name with a colon is to allow future implementations that support named labels for flow control. See the RATIONALE for *break* on page 97.

It is possible that a future version of this volume of IEEE Std. 1003.1-200x may require that '{' and '}' be treated individually as control operators, although the token "{ }" will probably be a special-case exemption from this because of the often-used *find{ }* construct.

2.5 Parameters and Variables

A parameter can be denoted by a name, a number, or one of the special characters listed in Section 2.5.2. A variable is a parameter denoted by a name.

A parameter is set if it has an assigned value (null is a valid value). Once a variable is set, it can only be unset by using the *unset* special built-in command.

2.5.1 Positional Parameters

A positional parameter is a parameter denoted by the decimal value represented by one or more digits, other than the single digit 0. The digits denoting the positional parameters are always interpreted as a decimal value, even if there is a leading zero. When a positional parameter with more than one digit is specified, the application shall enclose the digits in braces (see Section 2.6.2 on page 51). Positional parameters are initially assigned when the shell is invoked (see *sh*), temporarily replaced when a shell function is invoked (see Section 2.9.5 on page 79), and can be reassigned with the *set* special built-in command.

Rationale

The digits denoting the positional parameters are always interpreted as a decimal value, even if there is a leading zero.

2.5.2 Special Parameters

Listed below are the special parameters and the values to which they shall expand. Only the values of the special parameters are listed; see Section 2.6 on page 49 for a detailed summary of all the stages involved in expanding words.

- @ Expands to the positional parameters, starting from one. When the expansion occurs within double-quotes, and where field splitting (see Section 2.6.5 on page 58) is performed, each positional parameter expands as a separate field, with the provision that the expansion of the first parameter is still joined with the beginning part of the original word (assuming that the expanded parameter was embedded within a word), and the expansion of the last parameter is still joined with the last part of the original word. If there are no positional parameters, the expansion of '@' generates zero fields, even when '@' is double-quoted.
- * Expands to the positional parameters, starting from one. When the expansion occurs within a double-quoted string (see Section 2.2.3 on page 36), it expands to a single field with the value of each parameter separated by the first character of the *IFS* variable, or by a <space> character if *IFS* is unset. If *IFS* is set to a null string, this is not equivalent to unsetting it; its first character does not exist, so the parameter values are concatenated.
- # Expands to the decimal number of positional parameters. The command name (parameter 0) is not counted in the number given by '#' because it is a special parameter, not a positional parameter.
- ? Expands to the decimal exit status of the most recent pipeline (see Section 2.9.2 on page 72).
- (Hyphen.) Expands to the current option flags (the single-letter option names concatenated into a string) as specified on invocation by the *set* special built-in command or implicitly by the shell.
- \$ Expands to the decimal process ID of the invoked shell. In a subshell (see Section 2.12 on page 90), '\$' shall expand to the same value as that of the current shell.

1547 Most historical implementations implement subshells by forking; thus, the special
 1548 parameter '\$' does not necessarily represent the process ID of the shell process executing
 1549 the commands since the subshell execution environment preserves the value of '\$'.

1550 ! Expands to the decimal process ID of the most recent background command (see Section
 1551 2.9.3 on page 73) executed from the current shell. (For example, background commands
 1552 executed from subshells do not affect the value of "\$!" in the current shell environment.)
 1553 For a pipeline, the process ID is that of the last command in the pipeline.

1554 0 (Zero.) Expands to the name of the shell or shell script. See *sh* on page 888 for a detailed
 1555 description of how this name is derived.

1556 See the description of the *IFS* variable in Section 2.5.3 on page 45.

1557 Rationale

1558 Most historical implementations implement subshells by forking; thus, the special parameter
 1559 '\$' does not necessarily represent the process ID of the shell process executing the commands
 1560 since the subshell execution environment preserves the value of '\$'.

1561 If a subshell were to execute a background command, the value of "\$!" for the parent would
 1562 not change. For example:

```
1563 (
1564   date &
1565   echo $!
1566 )
1567 echo $!
```

1568 would echo two different values for "\$!".

1569 The "\$-" special parameter can be used to save and restore *set* options:

```
1570 Save=$(echo $- | sed 's/[ics]//g')
1571 ...
1572 set +aCefnuvx
1573 if [ -n "$Save" ]; then
1574     set -$Save
1575 fi
```

1576 The three options are removed using *sed* in the example because they may appear in the value of
 1577 "\$-" (from the *sh* command line), but are not valid options to *set*.

1578 The descriptions of parameters '*' and '@' assume the reader is familiar with the field splitting
 1579 discussion in Section 2.6.5 on page 58 and understands that portions of the word remain
 1580 concatenated unless there is some reason to split them into separate fields.

1581 Some examples of the '*' and '@' properties, including the concatenation aspects:

```
1582 set "abc" "def ghi" "jkl"
1583 echo $*      => "abc" "def" "ghi" "jkl"
1584 echo "$*"    => "abc def ghi jkl"
1585 echo $@      => "abc" "def" "ghi" "jkl"
```

1586 but:


```

1587      echo "$@"      => "abc" "def ghi" "jkl"
1588      echo "xx$@yy"   => "xxabc" "def ghi" "jkl yy"
1589      echo "$@$@"     => "abc" "def ghi" "jklabc" "def ghi" "jkl"

```

1590 In the preceding examples, the double-quote characters that appear after the "=>" do not appear
 1591 in the output and are used only to illustrate word boundaries.

1592 The following example illustrates the effect of setting *IFS* to a null string:

```

1593      $ IFS=' '
1594      $ set foo bar bam
1595      $ echo "$@"
1596      foo bar bam
1597      $ echo "$*"
1598      foobarbam
1599      $ unset IFS
1600      $ echo "$*"
1601      foo bar bam

```

1602 2.5.3 Shell Variables

1603 Variables shall be initialized from the environment (as defined by the System Interface
 1604 Definitions volume of IEEE Std. 1003.1-200x, Chapter 8, Environment Variables and the *exec*
 1605 function in the System Interfaces volume of IEEE Std. 1003.1-200x) and can be given new values
 1606 with variable assignment commands. If a variable is initialized from the environment, it shall be
 1607 marked for export immediately; see the *export* special built-in. New variables can be defined and
 1608 initialized with variable assignments, with the *read* or *getopts* utilities, with the *name* parameter in
 1609 a *for* loop, with the *\${name=word}* expansion, or with other mechanisms provided as
 1610 implementation extensions. The following variables shall affect the execution of the shell:

1611	<i>ENV</i>	This variable, when and only when an interactive shell is invoked, shall be
1612		subjected to parameter expansion (see Section 2.6.2 on page 51) by the shell
1613		and the resulting value shall be used as a path name of a file containing shell
1614		commands to execute in the current environment. The file need not be
1615		executable. If the expanded value of <i>ENV</i> is not an absolute path name, the
1616		results are unspecified. <i>ENV</i> shall be ignored if the user's real and effective
1617		user IDs or real and effective group IDs are different. This volume of
1618		IEEE Std. 1003.1-200x specifies the effects of this variable only for systems
1619		supporting the User Portability Utilities option.
1620	<i>HOME</i>	This variable shall be interpreted as the path name of the user's home
1621		directory. The contents of <i>HOME</i> are used in tilde expansion (see Section 2.6.1
1622		on page 50).
1623	<i>IFS</i>	(Input Field Separators.) A string treated as a list of characters that is used for
1624		field splitting and to split lines into fields with the <i>read</i> command. If <i>IFS</i> is not
1625		set, the shell shall behave as if the value of <i>IFS</i> were the <space>, <tab>, and
1626		<newline> characters; see Section 2.6.5 on page 58.
1627	<i>LANG</i>	This variable shall provide a default value for the internationalization
1628		variables that are unset or null. If <i>LANG</i> is unset or null, the corresponding
1629		value from the implementation-dependent default locale is used. If any of the
1630		internationalization variables contains an invalid setting, the utility behaves as
1631		if none of the variables had been defined.
1632	<i>LC_ALL</i>	This variable shall provide a default value for the <i>LC_*</i> variables, as described
1633		in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter

1634		8, Environment Variables.	
1635	<i>LC_COLLATE</i>	This variable shall determine the behavior of range expressions, equivalence	
1636		classes, and multi-character collating elements within pattern matching.	
1637	<i>LC_CTYPE</i>	This variable shall determine the interpretation of sequences of bytes of text	
1638		data as characters (for example, single-byte as opposed to multi-byte	
1639		characters), which characters are defined as letters (character class alpha) and	
1640		<blank> characters (character class blank), and the behavior of character	
1641		classes within pattern matching. Changing the value of <i>LC_CTYPE</i> after the	
1642		shell has started shall not affect the lexical processing of shell commands in	
1643		the current shell execution environment or its subshells. Invoking a shell	
1644		script or performing <i>exec sh</i> subjects the new shell to the changes in	
1645		<i>LC_CTYPE</i> .	
1646	<i>LC_MESSAGES</i>	This variable shall determine the language in which messages should be	
1647		written.	
1648	<i>LINENO</i>	This variable shall be set by the shell to a decimal number representing the	
1649		current sequential line number (numbered starting with 1) within a script or	
1650		function before it executes each command. If the user unsets or resets	
1651		<i>LINENO</i> , the variable may lose its special meaning for the life of the shell. If	
1652		the shell is not currently executing a script or function, the value of <i>LINENO</i> is	
1653		unspecified. This volume of IEEE Std. 1003.1-200x specifies the effects of the	
1654		variable only for systems supporting the User Portability Utilities option.	
1655	XSI <i>NLSPATH</i>	This variable shall determine the location of message catalogs for the	
1656		processing of <i>LC_MESSAGES</i> .	
1657	<i>PATH</i>	This variable represents a string formatted as described in the System	
1658		Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 8, Environment	
1659		Variables, used to effect command interpretation; see Section 2.9.1.1 on page	
1660		69.	
1661	<i>PPID</i>	This variable shall be set by the shell to the decimal process ID of the process	
1662		that invoked this shell. In a subshell (see Section 2.12 on page 90), <i>PPID</i> shall	
1663		be set to the same value as that of the parent of the current shell. For example,	
1664		<i>echo\$PPID</i> and (<i>echo\$PPID</i>) would produce the same value. This volume of	
1665		IEEE Std. 1003.1-200x specifies the effects of the variable only for systems	
1666		supporting the User Portability Utilities option.	
1667	<i>PS1</i>	Each time an interactive shell is ready to read a command, the value of this	
1668		variable shall be subjected to parameter expansion and written to standard	
1669		error. The default value shall be "\$ ". For users who have specific additional	
1670		implementation-dependent privileges, the default may be another,	
1671		implementation-dependent value. (Historically, the superuser has had a	
1672		prompt of '#'.) The shell shall replace each instance of the character '!' in	
1673		<i>PS1</i> with the history file number of the next command to be typed. Escaping	
1674		the '!' with another '!' (that is, "!!") shall place the literal character '!' in	
1675		the prompt. This volume of IEEE Std. 1003.1-200x specifies the effects of the	
1676		variable only for systems supporting the User Portability Utilities option.	
1677	<i>PS2</i>	Each time the user enters a <newline> character prior to completing a	
1678		command line in an interactive shell, the value of this variable shall be	
1679		subjected to parameter expansion and written to standard error. The default	
1680		value is "> ". This volume of IEEE Std. 1003.1-200x specifies the effects of the	
1681		variable only for systems supporting the User Portability Utilities option.	

1682	<i>PS4</i>	When an execution trace (<i>set -x</i>) is being performed in an interactive shell,
1683		before each line in the execution trace, the value of this variable shall be
1684		subjected to parameter expansion and written to standard error. The default
1685		value is "+ ". This volume of IEEE Std. 1003.1-200x specifies the effects of the
1686		variable only for systems supporting the User Portability Utilities option.
1687	<i>PWD</i>	This variable shall be set by the shell to be an absolute path name of the
1688		current working directory, containing no components of type symbolic link,
1689		no components that are dot, and no components that are dot-dot when the
1690		shell is initialized. If an application sets or unsets the value of <i>PWD</i> , the
1691		behaviors of the <i>cd</i> and <i>pwd</i> utilities are unspecified.
1692	Rationale	
1693		See the discussion of <i>IFS</i> in Rationale on page 58.
1694		The prohibition on <i>LC_CTYPE</i> changes affecting lexical processing protects the shell
1695		implementor (and the shell programmer) from the ill effects of changing the definition of
1696		<blank> or the set of alphabetic characters in the current environment. It would probably not be
1697		feasible to write a compiled version of a shell script without this rule. The rule applies only to
1698		the current invocation of the shell and its subshells—invoking a shell script or performing <i>exec sh</i>
1699		would subject the new shell to the changes in <i>LC_CTYPE</i> .
1700		Other common environment variables used by historical shells are not specified by this volume
1701		of IEEE Std. 1003.1-200x, but they should be reserved for the historical uses.
1702		Tilde expansion for components of the <i>PATH</i> in an assignment such as:
1703		<code>PATH=~hlj/bin:~dwc/bin:\$PATH</code>
1704		is a feature of some historical shells and is allowed by the wording of Section 2.6.1 on page 50.
1705		Note that the tildes are expanded during the assignment to <i>PATH</i> , not when <i>PATH</i> is accessed
1706		during command search.
1707		The following entries represent additional information about variables included in this volume
1708		of IEEE Std. 1003.1-200x, or rationale for common variables in use by shells that have been
1709		excluded:
1710	—	(Underscore.) While underscore is historical practice, its overloaded usage in
1711		the KornShell is confusing, and it has been omitted from this volume of
1712		IEEE Std. 1003.1-200x.
1713	<i>ENV</i>	This variable can be used to set aliases and other items local to the invocation
1714		of a shell. The file referred to by <i>ENV</i> differs from <i>\$HOME/.profile</i> in that
1715		<i>.profile</i> is typically executed at session start-up, whereas the <i>ENV</i> file is
1716		executed at the beginning of each shell invocation. The <i>ENV</i> value is
1717		interpreted in a manner similar to a dot script, in that the commands are
1718		executed in the current environment and the file needs to be readable, but not
1719		executable. However, unlike dot scripts, no <i>PATH</i> searching is performed.
1720		This is used as a guard against Trojan Horse security breaches.
1721	<i>ERRNO</i>	This variable was omitted from this volume of IEEE Std. 1003.1-200x because
1722		the values of error numbers are not defined in the System Interface Definitions
1723		volume of IEEE Std. 1003.1-200x in a portable manner.
1724	<i>FCEDIT</i>	Since this variable affects only the <i>fc</i> utility, it has been omitted from this more
1725		global place. The value of <i>FCEDIT</i> does not affect the command line editing
1726		mode in the shell; see the description of <i>set -o vi</i> in <i>set</i> on page 117.

1727	<i>PS1</i>	This variable is used for interactive prompts. Historically, the “superuser”
1728		has had a prompt of ‘#’. Since privileges are not required to be monolithic, it
1729		is difficult to define which privileges should cause the alternate prompt.
1730		However, a sufficiently powerful user should be reminded of that power by
1731		having an alternate prompt.
1732	<i>PS3</i>	This variable is used by the KornShell for the <i>select</i> command. Since the POSIX
1733		shell does not include <i>select</i> , <i>PS3</i> was omitted.
1734	<i>PS4</i>	This variable is used for shell debugging. For example, the following script:
1735		<pre>PS4='[\${LINENO}]+ '</pre>
1736		<pre>set -x</pre>
1737		<pre>echo Hello</pre>
1738		writes the following to standard error:
1739		<pre>[3]+ echo Hello</pre>
1740	<i>RANDOM</i>	This pseudo-random number generator was not seen as being useful to
1741		interactive users.
1742	<i>SECONDS</i>	Although this variable is sometimes used with <i>PS1</i> to allow the display of the
1743		current time in the prompt of the user, it is not one that would be manipulated
1744		frequently enough by an interactive user to include in this volume of
1745		IEEE Std. 1003.1-200x.

2.6 Word Expansions

This section describes the various expansions that are performed on words. Not all expansions are performed on every word, as explained in the following sections.

Tilde expansions, parameter expansions, command substitutions, arithmetic expansions, and quote removals that occur within a single word expand to a single field. It is only field splitting or path name expansion that can create multiple fields from a single word. The single exception to this rule is the expansion of the special parameter '@' within double-quotes, as described in Section 2.5.2 on page 43.

The order of word expansion shall be as follows:

1. Tilde expansion (see Section 2.6.1 on page 50), parameter expansion (see Section 2.6.2 on page 51), command substitution (see Section 2.6.3 on page 54), and arithmetic expansion (see Section 2.6.4 on page 56) shall be performed, beginning to end. See item 5 in Section 2.3 on page 39.
2. Field splitting (see Section 2.6.5 on page 58) shall be performed on the portions of the fields generated by step 1, unless *IFS* is null.
3. Path name expansion (see Section 2.6.6 on page 59) shall be performed, unless *set -f* is in effect.
4. Quote removal (see Section 2.6.7 on page 59) shall always be performed last.

The expansions described in this section shall occur in the same shell environment as that in which the command is executed.

If the complete expansion appropriate for a word results in an empty field, that empty field shall be deleted from the list of fields that form the completely expanded command, unless the original word contained single-quote or double-quote characters.

The '\$' character is used to introduce parameter expansion, command substitution, or arithmetic evaluation. If an unquoted '\$' is followed by a character that is either not numeric, the name of one of the special parameters (see Section 2.5.2 on page 43), a valid first character of a variable name, a left curly brace ('{') or a left parenthesis, the result is unspecified.

Rationale

Step (2) refers to the “portions of fields generated by step (1)”. For example, if the word being expanded were "\$x+\$y" and *IFS*=+, the word would be split only if "\$x" or "\$y" contained '+'; the '+' in the original word was not generated by step (1).

IFS is used for performing field splitting on the results of parameter and command substitution; it is not used for splitting all fields. Previous versions of the shell used it for splitting all fields during field splitting, but this has severe problems because the shell can no longer parse its own script. There are also important security implications caused by this behavior. All useful applications of *IFS* use it for parsing input of the *read* utility and for splitting the results of parameter and command substitution.

The rule concerning expansion to a single field requires that if **foo=abc** and **bar=def**, that:

```
"$foo" "$bar"
```

expands to the single field:

```
abcdef
```

The rule concerning empty fields can be illustrated by:

```
$ unset foo
$ set $foo bar ' ' xyz "$foo" abc
$ for i
> do
>     echo "-$i-"
> done
-bar-
—
-xyz-
—
-abc-
```

Step (1) indicates that parameter expansion, command substitution, and arithmetic expansion are all processed simultaneously as they are scanned. For example, the following is valid arithmetic:

```
x=1
echo $(( $(echo 3)+$x ))
```

An early proposal stated that tilde expansion preceded the other steps, but this is not the case in known historical implementations; if it were, and if a referenced home directory contained a '\$' character, expansions would result within the directory name.

2.6.1 Tilde Expansion

A *tilde-prefix* consists of an unquoted tilde character at the beginning of a word, followed by all of the characters preceding the first unquoted slash in the word, or all the characters in the word if there is no slash. In an assignment (see the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 3.426, Variable Assignment), multiple tilde-prefixes can be used: at the beginning of the word (that is, following the equal sign of the assignment), following any unquoted colon, or both. A tilde-prefix in an assignment is terminated by the first unquoted colon or slash. If none of the characters in the tilde-prefix are quoted, the characters in the tilde-prefix following the tilde are treated as a possible login name from the user database. A portable login name cannot contain characters outside the set given in the description of the *LOGNAME* environment variable in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 8.3, Other Environment Variables. If the login name is null (that is, the tilde-prefix contains only the tilde), the tilde-prefix is replaced by the value of the variable *HOME*. If *HOME* is unset, the results are unspecified. Otherwise, the tilde-prefix is replaced by a path name of the initial working directory associated with the login name obtained using the *getpwnam()* function as defined in the System Interfaces volume of IEEE Std. 1003.1-200x. If the system does not recognize the login name, the results are undefined.

Rationale

Tilde expansion generally occurs only at the beginning of words, but an exception based on historical practice has been included:

```
PATH=/posix/bin:~dkg/bin
```

This is eligible for tilde expansion because tilde follows a colon and none of the relevant characters is quoted. Consideration was given to prohibiting this behavior because any of the following are reasonable substitutes:

```

1831     PATH=$(printf %s ~karels/bin : ~bostic/bin)
1832     for Dir in ~maat/bin ~srb/bin ...
1833     do
1834         PATH=${PATH:+$PATH:}$Dir
1835     done

```

1836 In the first command, explicit colons are used for each directory. In all cases, the shell performs
 1837 tilde expansion on each directory because all are separate words to the shell.

1838 Note that expressions in operands such as:

```

1839     make -k mumble LIBDIR=~chet/lib

```

1840 do not qualify as shell variable assignments and tilde expansion is not performed (unless the
 1841 command does so itself, which *make* does not).

1842 Because of the requirement that the word is not quoted, the following are not equivalent; only
 1843 the last causes tilde expansion:

```

1844     \~hlj/    ~h\lj/    ~"hlj"/    ~hlj\ /    ~hlj/

```

1845 In an early proposal, tilde expansion occurred following any unquoted equals sign or colon, but
 1846 this was removed because of its complexity and to avoid breaking commands such as:

```

1847     rcp hostname:~marc/.profile .

```

1848 A suggestion was made that the special sequence "\$~" should be allowed to force tilde
 1849 expansion anywhere. Since this is not historical practice, it has been left for future
 1850 implementations to evaluate. (The description in Section 2.2 on page 36 requires that a dollar
 1851 sign be quoted to represent itself, so the "\$~" combination is already unspecified.)

1852 The results of giving tilde with an unknown login name are undefined because the KornShell
 1853 "~+" and "~-" constructs make use of this condition, but in general it is an error to give an
 1854 incorrect login name with tilde. The results of having *HOME* unset are unspecified because some
 1855 historical shells treat this as an error.

1856 2.6.2 Parameter Expansion

1857 The format for parameter expansion is as follows:

```

1858     ${expression}

```

1859 where *expression* consists of all characters until the matching '}'. Any '}' escaped by a
 1860 backslash or within a quoted string, and characters in embedded arithmetic expansions,
 1861 command substitutions, and variable expansions, shall not be examined in determining the
 1862 matching '}'.

1863 The simplest form for parameter expansion is:

```

1864     ${parameter}

```

1865 The value, if any, of *parameter* shall be substituted.

1866 The parameter name or symbol can be enclosed in braces, which are optional except for
 1867 positional parameters with more than one digit or when *parameter* is followed by a character that
 1868 could be interpreted as part of the name. The matching closing brace shall be determined by
 1869 counting brace levels, skipping over enclosed quoted strings, and command substitutions.

1870 If the parameter name or symbol is not enclosed in braces, the expansion shall use the longest
 1871 valid name (see the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 3.234,
 1872 Name), whether or not the symbol represented by that name exists.

If a parameter expansion occurs inside double-quotes:

- Path name expansion shall not be performed on the results of the expansion.
- Field splitting shall not be performed on the results of the expansion, with the exception of '@'; see Section 2.5.2 on page 43.

In addition, a parameter expansion can be modified by using one of the following formats. In each case that a value of *word* is needed (based on the state of *parameter*, as described below), *word* shall be subjected to tilde expansion, parameter expansion, command substitution, and arithmetic expansion. If *word* is not needed, it shall not be expanded. The '}' character that delimits the following parameter expansion modifications shall be determined as described previously in this section and in Section 2.2.3 on page 36. (For example, \${foo-bar}xyz would result in the expansion of foo followed by the string xyz if foo is set, else the string "barxyz").

Use Default Values. If *parameter* is unset or null, the expansion of *word* shall be substituted; otherwise, the value of *parameter* shall be substituted.

Assign Default Values. If *parameter* is unset or null, the expansion of *word* shall be assigned to *parameter*. In all cases, the final value of *parameter* shall be substituted. Only variables, not positional parameters or special parameters, can be assigned in this way.

Indicate Error if Null or Unset. If *parameter* is unset or null, the expansion of *word* (or a message indicating it is unset if *word* is omitted) shall be written to standard error and the shell exits with a non-zero exit status. Otherwise, the value of *parameter* shall be substituted. An interactive shell need not exit.

Use Alternative Value. If *parameter* is unset or null, null shall be substituted; otherwise, the expansion of *word* shall be substituted.

In the parameter expansions shown previously, use of the colon in the format results in a test for a parameter that is unset or null; omission of the colon results in a test for a parameter that is only unset. The following table summarizes the effect of the colon:

	<i>parameter</i> Set and Not Null	<i>parameter</i> Set But Null	<i>parameter</i> Unset
<i>\${parameter:-word}</i>	substitute <i>parameter</i>	substitute <i>word</i>	substitute <i>word</i>
<i>\${parameter-word}</i>	substitute <i>parameter</i>	substitute null	substitute <i>word</i>
<i>\${parameter:=word}</i>	substitute <i>parameter</i>	assign <i>word</i>	assign <i>word</i>
<i>\${parameter=word}</i>	substitute <i>parameter</i>	substitute <i>parameter</i>	assign null
<i>\${parameter:?word}</i>	substitute <i>parameter</i>	error, exit	error, exit
<i>\${parameter?word}</i>	substitute <i>parameter</i>	substitute null	error, exit
<i>\${parameter:+word}</i>	substitute <i>word</i>	substitute null	substitute null
<i>\${parameter+word}</i>	substitute <i>word</i>	substitute <i>word</i>	substitute null

In all cases shown with “substitute”, the expression is replaced with the value shown. In all cases shown with “assign”, *parameter* is assigned that value, which also replaces the expression.

String Length. The length in characters of the value of *parameter* shall be substituted. If *parameter* is '*' or '@', the result of the expansion is unspecified.

The following four varieties of parameter expansion provide for substring processing. In each case, pattern matching notation (see Section 2.13 on page 92), rather than regular expression notation, shall be used to evaluate the patterns. If *parameter* is '*' or '@', the result of the

1919 expansion is unspecified. Enclosing the full parameter expansion string in double-quotes shall
 1920 not cause the following four varieties of pattern characters to be quoted, whereas quoting
 1921 characters within the braces shall have this effect.

1922 **`${parameter%word}`** **Remove Smallest Suffix Pattern.** The *word* is expanded to produce a
 1923 pattern. The parameter expansion then results in *parameter*, with the
 1924 smallest portion of the suffix matched by the *pattern* deleted.

1925 **`${parameter%%word}`** **Remove Largest Suffix Pattern.** The *word* shall be expanded to produce a
 1926 pattern. The parameter expansion then results in *parameter*, with the
 1927 largest portion of the suffix matched by the *pattern* deleted.

1928 **`${parameter#word}`** **Remove Smallest Prefix Pattern.** The *word* shall be expanded to produce
 1929 a pattern. The parameter expansion then results in *parameter*, with the
 1930 smallest portion of the prefix matched by the *pattern* deleted.

1931 **`${parameter##word}`** **Remove Largest Prefix Pattern.** The *word* shall be expanded to produce a
 1932 pattern. The parameter expansion then results in *parameter*, with the
 1933 largest portion of the prefix matched by the *pattern* deleted.

1934 Examples

1935 **`${parameter:-word}`**
 1936 In this example, *ls* is executed only if *x* is null or unset. (The `$(ls)` command substitution
 1937 notation is explained in Section 2.6.3 on page 54.)

1938 `$ {x:-$(ls) }`

1939 **`${parameter:=word}`**
 1940 `unset X`
 1941 `echo ${X:=abc}`
 1942 **`abc`**

1943 **`${parameter:?word}`**
 1944 `unset posix`
 1945 `echo ${posix:?}`
 1946 **`sh: posix: parameter null or not set`**

1947 **`${parameter:+word}`**
 1948 `set a b c`
 1949 `echo ${3:+posix}`
 1950 **`posix`**

1951 **`${#parameter}`**
 1952 `HOME=/usr/posix`
 1953 `echo ${#HOME}`
 1954 **`10`**

1955 **`${parameter%word}`**
 1956 `x=file.c`
 1957 `echo ${x%.c}.o`
 1958 **`file.o`**

1959 **`${parameter%%word}`**
 1960 `x=posix/src/std`
 1961 `echo ${x%%/*}`
 1962 **`posix`**

```

1963     ${parameter#word}
1964         x=$HOME/src/cmd
1965         echo ${x#$HOME}
1966         /src/cmd

```

```

1967     ${parameter##word}
1968         x=/one/two/three
1969         echo ${x##*/}
1970         three

```

1971 The double-quoting of patterns is different depending on where the double-quotes are placed:

1972 The asterisk is a pattern character.

1973 \${x# The literal asterisk is quoted and not special.

1974 **Rationale**

1975 The rule for finding the closing '}' in "\${...}" is the one used in the KornShell and is
 1976 upwardly compatible with the Bourne shell, which does not determine the closing '}' until the
 1977 word is expanded. The advantage of this is that incomplete expansions, such as:

```
1978     ${foo
```

1979 can be determined during tokenization, rather than during expansion.

1980 The string length and substring capabilities were included because of the demonstrated need for
 1981 them, based on their usage in other shells, such as C shell and KornShell.

1982 Historical versions of the KornShell have not performed tilde expansion on the word part of
 1983 parameter expansion; however, it is more consistent to do so.

1984 **2.6.3 Command Substitution**

1985 Command substitution allows the output of a command to be substituted in place of the
 1986 command name itself. Command substitution shall occur when the command is enclosed as
 1987 follows:

```
1988     $( command)
```

1989 or (backquoted version):

```
1990     `command`
```

1991 The shell shall expand the command substitution by executing *command* in a subshell
 1992 environment (see Section 2.12 on page 90) and replacing the command substitution (the text of
 1993 *command* plus the enclosing "\$()" or backquotes) with the standard output of the command,
 1994 removing sequences of one or more <newline> characters at the end of the substitution.
 1995 Embedded <newline> characters before the end of the output shall not be removed; however,
 1996 they may be treated as field delimiters and eliminated during field splitting, depending on the
 1997 value of *IFS* and quoting that is in effect.

1998 Within the backquoted style of command substitution, backslash shall retain its literal meaning,
 1999 except when followed by: '\$', '`', or '\`' (dollar sign, backquote, backslash). The search for
 2000 the matching backquote shall be satisfied by the first backquote found without a preceding
 2001 backslash; during this search, if a non-escaped backquote is encountered within a shell
 2002 comment, a here-document, an embedded command substitution of the \$(*command*) form, or a
 2003 quoted string, undefined results occur. A single-quoted or double-quoted string that begins, but
 2004 does not end, within the "`...`" sequence produces undefined results.

With the `$ (command)` form, all characters following the open parenthesis to the matching closing parenthesis constitute the *command*. Any valid shell script can be used for *command*, except:

- A script consisting solely of redirections produces unspecified results
- See the restriction on single subshells described below

The results of command substitution shall not be processed for further tilde expansion, parameter expansion, command substitution, or arithmetic expansion. If a command substitution occurs inside double-quotes, it shall not be performed on the results of the substitution.

Command substitution can be nested. To specify nesting within the backquoted version, the application shall precede the inner backquotes with backslashes, for example:

```
\`command\`
```

If the command substitution consists of a single subshell, such as:

```
$( (command) )
```

a portable application shall separate the `"$("` and `'('` into two tokens (that is, separate them with white space). This is required to avoid any ambiguities with arithmetic expansion.

Rationale

The `"$()"` form of command substitution solves a problem of inconsistent behavior when using backquotes. For example:

Command	Output
<code>echo '\\$x'</code>	<code>\\$x</code>
<code>echo `echo '\\$x'`</code>	<code>\$x</code>
<code>echo \$(echo '\\$x')</code>	<code>\\$x</code>

Additionally, the backquoted syntax has historical restrictions on the contents of the embedded command. While the newer `"$()"` form can process any kind of valid embedded script, the backquoted form cannot handle some valid scripts that include backquotes. For example, these otherwise valid embedded scripts do not work in the left column, but do work on the right:

<code>echo `</code>	<code>echo \$(</code>
<code>cat <<\eof</code>	<code>cat <<\eof</code>
<code>a here-doc with `</code>	<code>a here-doc with)</code>
<code>eof</code>	<code>eof</code>
<code>`</code>	<code>)</code>
<code>echo `</code>	<code>echo \$(</code>
<code>echo abc # a comment with `</code>	<code>echo abc # a comment with)</code>
<code>`</code>	<code>)</code>
<code>echo `</code>	<code>echo \$(</code>
<code>echo ` ` `</code>	<code>echo ` ` `</code>
<code>`</code>	<code>)</code>

Because of these inconsistent behaviors, the backquoted variety of command substitution is not recommended for new applications that nest command substitutions or attempt to embed complex scripts.

The KornShell feature:

If *command* is of the form *<word>*, *word* is expanded to generate a path name, and the value of the command substitution is the contents of this file with any trailing *<newline>*s deleted.

was omitted from this volume of IEEE Std. 1003.1-200x because *\$(cat word)* is an appropriate substitute. However, to prevent breaking numerous scripts relying on this feature, it is unspecified to have a script within "*\$ ()*" that has only redirections.

The requirement to separate "*\$ (*" and "*' (*" when a single subshell is command-substituted is to avoid any ambiguities with arithmetic expansion.

2.6.4 Arithmetic Expansion

Arithmetic expansion provides a mechanism for evaluating an arithmetic expression and substituting its value. The format for arithmetic expansion shall be as follows:

```
$((expression))
```

The expression shall be treated as if it were in double-quotes, except that a double-quote inside the expression is not treated specially. The shell expands all tokens in the expression for parameter expansion, command substitution, and quote removal.

Next, the shell shall treat this as an arithmetic expression and substitutes the value of the expression. The arithmetic expression shall be processed according to the rules of the ISO C standard, with the following exceptions:

- Only integer arithmetic is required.
- The *sizeof()* operator and the prefix and postfix "*++*" and "*—*" operators are not required.
- Selection, iteration, and jump statements are not supported.

As an extension, the shell may recognize arithmetic expressions beyond those listed. If the expression is invalid, the expansion fails and the shell shall write a message to standard error indicating the failure.

Examples

A simple example using arithmetic expansion:

```
# repeat a command 100 times
x=100
while [ $x -gt 0 ]
do
    command
    x=$((x-1))
done
```

Rationale

The "*(())*" form of KornShell arithmetic in early proposals was omitted. The standard developers concluded that there was a strong desire for some kind of arithmetic evaluator to replace *expr*, and that relating it to '*\$*' makes it work well with the standard shell language, and it provides access to arithmetic evaluation in places where accessing a utility would be inconvenient.

The syntax and semantics for arithmetic were changed for the ISO/IEC 9945-2:1993 standard. The language is essentially a pure arithmetic evaluator of constants and operators (excluding assignment) and represents a simple subset of the previous arithmetic language (which was derived from the KornShell "*(())*" construct). The syntax was changed from that of a

command denoted by `((expression))` to an expansion denoted by `$((expression))`. The new form is a dollar expansion (`'$'`) that evaluates the expression and substitutes the resulting value. Objections to the previous style of arithmetic included that it was too complicated, did not fit in well with the use of variables in the shell, and its syntax conflicted with subshells. The justification for the new syntax is that the shell is traditionally a macro language, and if a new feature is to be added, it should be accomplished by extending the capabilities presented by the current model of the shell, rather than by inventing a new one outside the model; adding a new dollar expansion was perceived to be the most intuitive and least destructive way to add such a new capability.

In early proposals, a form `$[expression]` was used. It was functionally equivalent to the `"$(())"` of the current text, but objections were lodged that the 1988 KornShell had already implemented `"$(())"` and there was no compelling reason to invent yet another syntax. Furthermore, the `"$["` syntax had a minor incompatibility involving the patterns in `case` statements.

The portion of the ISO C standard arithmetic operations selected corresponds to the operations historically supported in the KornShell.

It was concluded that the `test` command (`()`) was sufficient for the majority of relational arithmetic tests, and that tests involving complicated relational expressions within the shell are rare, yet could still be accommodated by testing the value of `"$(())"` itself. For example:

```
# a complicated relational expression
while [ $(( ($x + $y)/($a * $b)) < ($foo*$bar) )) -ne 0 ]
```

or better yet, the rare script that has many complex relational expressions could define a function like this:

```
val() {
    return $(!$1)
}
```

and complicated tests would be less intimidating:

```
while val $(( ($x + $y)/($a * $b)) < ($foo*$bar) ))
do
    # some calculations
done
```

A suggestion that was not adopted was to modify *true* and *false* to take an optional argument, and *true* would exit true only if the argument was non-zero, and *false* would exit false only if the argument was non-zero:

```
while true $(( $x > 5 && $y <= 25 ))
```

There is a minor portability concern with the new syntax. The example `$((2+2))` could have been intended to mean a command substitution of a utility named `2+2` in a subshell. The standard developers considered this to be obscure and isolated to some KornShell scripts (because `"$(())"` command substitution existed previously only in the KornShell). The text on command substitution requires that the `"$("` and `'('` be separate tokens if this usage is needed.

An example such as:

```
echo $((echo hi);(echo there))
```

should not be misinterpreted by the shell as arithmetic because attempts to balance the parentheses pairs would indicate that they are subshells. However, as indicated by the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 3.115, Control Operator, a conforming application must separate two adjacent parentheses with white space to indicate

2133 nested subshells.

2134 2.6.5 Field Splitting

2135 After parameter expansion (Section 2.6.2 on page 51), command substitution (Section 2.6.3 on
2136 page 54), and arithmetic expansion (Section 2.6.4 on page 56), the shell shall scan the results of
2137 expansions and substitutions that did not occur in double-quotes for field splitting and multiple
2138 fields can result.

2139 The shell shall treat each character of the *IFS* as a delimiter and uses the delimiters to split the
2140 results of parameter expansion and command substitution into fields.

- 2141 1. If the value of *IFS* is a <space>, <tab>, and <newline> character, or if it is unset, any
2142 sequence of <space>, <tab>, or <newline> characters at the beginning or end of the input
2143 shall be ignored and any sequence of those characters within the input shall delimit a field.
2144 For example, the input:

2145 `<newline><space><tab>foo<tab><tab>bar<space>`

2146 yields two fields, **foo** and **bar**.

- 2147 2. If the value of *IFS* is null, no field splitting shall be performed.
- 2148 3. Otherwise, the following rules shall be applied in sequence. The term “*IFS* white space” is
2149 used to mean any sequence (zero or more instances) of white space characters that are in
2150 the *IFS* value (for example, if *IFS* contains <space>/<comma>/<tab>, any sequence of
2151 <space> and <tab> characters is considered *IFS* white space).
 - 2152 a. *IFS* white space shall be ignored at the beginning and end of the input.
 - 2153 b. Each occurrence in the input of an *IFS* character that is not *IFS* white space, along
2154 with any adjacent *IFS* white space, shall delimit a field, as described previously.
 - 2155 c. Non-zero-length *IFS* white space shall delimit a field.

2156 Rationale

2157 The operation of field splitting using *IFS*, as described in early proposals, was based on the way
2158 the KornShell splits words, but it is incompatible with other common versions of the shell.
2159 However, each has merit, and so a decision was made to allow both. If the *IFS* variable is unset
2160 or is <space><tab><newline>, the operation is equivalent to the way the System V shell splits
2161 words. Using characters outside the <space><tab><newline> set yields the KornShell behavior,
2162 where each of the non-<space><tab><newline> characters is significant. This behavior, which
2163 affords the most flexibility, was taken from the way the original *awk* handled field splitting.

2164 Rule (3) can be summarized as a pseudo-ERE:

2165 $(s^*ns^* | s^+)$

2166 where *s* is an *IFS* white space character and *n* is a character in the *IFS* that is not white space.
2167 Any string matching that ERE delimits a field, except that the *s*⁺ form does not delimit fields at
2168 the beginning or the end of a line. For example, if *IFS* is <space>/<comma>/<tab>, the string:

2169 `<space><space>red<space><space> , <space>white<space>blue`

2170 yields the three colors as the delimited fields.

2171 2.6.6 Path Name Expansion

2172 After field splitting, if *set -f* is not in effect, each field in the resulting command line shall be
2173 expanded using the algorithm described in Section 2.13 on page 92, qualified by the rules in
2174 Section 2.13.3 on page 94.

2175 2.6.7 Quote Removal

2176 The quote characters: `'\'`, `'''`, and `''` (backslash, single-quote, double-quote) that were
2177 present in the original word shall be removed unless they have themselves been quoted.

2178 2.7 Redirection

2179 Redirection is used to open and close files for the current shell execution environment (see
 2180 Section 2.12 on page 90) or for any command. *Redirection operators* can be used with numbers
 2181 representing file descriptors (see the System Interface Definitions volume of
 2182 IEEE Std. 1003.1-200x, Section 3.169, File Descriptor) as described below.

2183 The overall format used for redirection is:

2184 `[n]redir-op word`

2185 The number *n* is an optional decimal number designating the file descriptor number; the
 2186 application shall ensure it is delimited from any preceding text and immediately precede the
 2187 redirection operator *redir-op*. If *n* is quoted, the number shall not be recognized as part of the
 2188 redirection expression. For example:

2189 `echo \2>a`

2190 writes the character 2 into file **a**. If any part of *redir-op* is quoted, no redirection expression is
 2191 recognized. For example:

2192 `echo 2\>a`

2193 writes the characters 2>a to standard output. The optional number, redirection operator, and
 2194 *word* shall not appear in the arguments provided to the command to be executed (if any).

2195 Open files are represented by decimal numbers starting with zero. The largest possible value is
 2196 implementation-dependent; however, all implementations support at least 0 to 9, inclusive, for
 2197 use by the application. These numbers are called *file descriptors*. The values 0, 1, and 2 have
 2198 special meaning and conventional uses and are implied by certain redirection operations; they
 2199 are referred to as *standard input*, *standard output*, and *standard error*, respectively. Programs
 2200 usually take their input from standard input, and write output on standard output. Error
 2201 messages are usually written on standard error. The redirection operators can be preceded by
 2202 one or more digits (with no intervening <blank> characters allowed) to designate the file
 2203 descriptor number.

2204 If the redirection operator is "<<" or "<<-", the word that follows the redirection operator shall
 2205 be subjected to quote removal; it is unspecified whether any of the other expansions occur. For
 2206 the other redirection operators, the word that follows the redirection operator shall be subjected
 2207 to tilde expansion, parameter expansion, command substitution, arithmetic expansion, and
 2208 quote removal. Path name expansion shall not be performed on the word by a non-interactive
 2209 shell; an interactive shell may perform it, but does do so only when the expansion would result
 2210 in one word.

2211 If more than one redirection operator is specified with a command, the order of evaluation is
 2212 from beginning to end.

2213 A failure to open or create a file shall cause a redirection to fail.

2214 2.7.1 Redirecting Input

2215 Input redirection shall cause the file whose name results from the expansion of *word* to be
 2216 opened for reading on the designated file descriptor, or standard input if the file descriptor is not
 2217 specified.

2218 The general format for redirecting input is:

2219 `[n]<word`

2220 where the optional *n* represents the file descriptor number. If the number is omitted, the
 2221 redirection shall refer to standard input (file descriptor 0).

2222 2.7.2 Redirecting Output

2223 The two general formats for redirecting output are:

2224 `[n]>word`

2225 `[n]>|word`

2226 where the optional *n* represents the file descriptor number. If the number is omitted, the
 2227 redirection shall refer to standard output (file descriptor 1).

2228 Output redirection using the '*>*' format fails if the *noclobber* option is set (see the description of
 2229 *set -C*) and the file named by the expansion of *word* exists and is a regular file. Otherwise,
 2230 redirection using the '*>*' or '*>|*' formats shall cause the file whose name results from the
 2231 expansion of *word* to be created and opened for output on the designated file descriptor, or
 2232 standard output if none is specified. If the file does not exist, it shall be created; otherwise, it
 2233 shall be truncated to be an empty file after being opened.

2234 2.7.3 Appending Redirected Output

2235 Appended output redirection shall cause the file whose name results from the expansion of
 2236 *word* to be opened for output on the designated file descriptor. The file is opened as if the *open()*
 2237 function as defined in the System Interfaces volume of IEEE Std. 1003.1-200x was called with the
 2238 *O_APPEND* flag. If the file does not exist, it shall be created.

2239 The general format for appending redirected output is as follows:

2240 `[n]>>word`

2241 where the optional *n* represents the file descriptor number. If the number is omitted, the
 2242 redirection refers to standard output (file descriptor 1).

2243 2.7.4 Here-Document

2244 The redirection operators "*<<*" and "*<<-*" both allow redirection of lines contained in a shell
 2245 input file, known as a *here-document*, to the input of a command.

2246 The here-document shall be treated as a single word that begins after the next *<newline>*
 2247 character and continues until there is a line containing only the delimiter, with no trailing
 2248 *<blank>* characters. Then the next here-document starts, if there is one. The format is as follows:

2249 `[n]<<word`

2250 `here-document`

2251 `delimiter`

2252 where the optional *n* represents the file descriptor number. If the number is omitted, the here-
 2253 document refers to standard output (file descriptor 0).

2254 If any character in *word* is quoted, the delimiter shall be formed by performing quote removal on
 2255 *word*, and the here-document lines are not expanded. Otherwise, the delimiter shall be the *word*
 2256 itself.

2257 If no characters in *word* are quoted, all lines of the here-document shall be expanded for
 2258 parameter expansion, command substitution, and arithmetic expansion. In this case, the
 2259 backslash in the input behaves as the backslash inside double-quotes (see Section 2.2.3 on page
 2260 36). However, the double-quote character (') ' shall not be treated specially within a here-
 2261 document, except when the double-quote appears within "\$ () ", " ` ` ", or "\$ { } ".

2262 If the redirection symbol is "<<-", all leading tab characters shall be stripped from input lines
 2263 and the line containing the trailing delimiter. If more than one "<<" or "<<-" operator is
 2264 specified on a line, the here-document associated with the first operator shall be supplied first by
 2265 the application and shall be read first by the shell.

2266 Examples

2267 An example of a here-document follows:

```
2268 cat <<eof1; cat <<eof2
2269 Hi ,
2270 eof1
2271 Helene.
2272 eof2
```

2273 2.7.5 Duplicating an Input File Descriptor

2274 The redirection operator:

```
2275 [n]<&word
```

2276 is used to duplicate one input file descriptor from another, or to close one. If *word* evaluates to
 2277 one or more digits, the file descriptor denoted by *n*, or standard input if *n* is not specified, shall
 2278 be made to be a copy of the file descriptor denoted by *word*; if the digits in *word* do not represent
 2279 a file descriptor already open for input, a redirection error shall result; see Section 2.8.1 on page
 2280 65. If *word* evaluates to '—', file descriptor *n*, or standard input if *n* is not specified, shall be
 2281 closed. If *word* evaluates to something else, the behavior is unspecified.

2282 2.7.6 Duplicating an Output File Descriptor

2283 The redirection operator:

```
2284 [n]>&word
```

2285 is used to duplicate one output file descriptor from another, or to close one. If *word* evaluates to
 2286 one or more digits, the file descriptor denoted by *n*, or standard output if *n* is not specified, shall
 2287 be made to be a copy of the file descriptor denoted by *word*; if the digits in *word* do not represent
 2288 a file descriptor already open for output, a redirection error shall result; see Section 2.8.1 on page
 2289 65. If *word* evaluates to '—', file descriptor *n*, or standard output if *n* is not specified, is closed. If
 2290 *word* evaluates to something else, the behavior is unspecified.

2.7.7 Open File Descriptors for Reading and Writing

The redirection operator:

```
[n]<>word
```

shall cause the file whose name is the expansion of *word* to be opened for both reading and writing on the file descriptor denoted by *n*, or standard input if *n* is not specified. If the file does not exist, it shall be created.

Rationale

In the System Interfaces volume of IEEE Std. 1003.1-200x, file descriptors are integers in the range 0–({OPEN_MAX}–1). The file descriptors discussed in Section 2.7 on page 60 are that same set of small integers.

Having multi-digit file descriptor numbers for I/O redirection can cause some obscure compatibility problems. Specifically, scripts that depend on an example command:

```
echo 22>/dev/null
```

echoing 2 to standard error or 22 to standard output are no longer portable. However, the file descriptor number still must be delimited from the preceding text. For example:

```
cat file2>foo
```

writes the contents of **file2**, not the contents of **file**.

The ">|" format of output redirection was adopted from the KornShell. Along with the *noclobber* option, *set -C*, it provides a safety feature to prevent inadvertent overwriting of existing files. (See the RATIONALE for *pathchk* on page 732 for why this step was taken.) The restriction on regular files is historical practice.

The System V shell and the KornShell have differed historically on path name expansion of *word*; the former never performed it, the latter only when the result was a single field (file). As a compromise, it was decided that the KornShell functionality was useful, but only as a shorthand device for interactive users. No reasonable shell script would be written with a command such as:

```
cat foo > a*
```

Thus, shell scripts are prohibited from doing it, while interactive users can select the shell with which they are most comfortable.

The construct 2>&1 is often used to redirect standard error to the same file as standard output. Since the redirections take place beginning to end, the order of redirections is significant. For example:

```
ls > foo 2>&1
```

directs both standard output and standard error to file **foo**. However:

```
ls 2>&1 > foo
```

only directs standard output to file **foo** because standard error was duplicated as standard output before standard output was directed to file **foo**.

The "<>" operator could be useful in writing an application that worked with several terminals, and occasionally wanted to start up a shell. That shell would in turn be unable to run applications that run from an ordinary controlling terminal unless it could make use of "<>" redirection. The specific example is a historical version of the pager *more*, which reads from

2332 standard error to get its commands, so standard input and standard output are both available
2333 for their usual usage. There is no way of saying the following in the shell without "<>":

```
2334     cat food | more - >/dev/tty03 2<>/dev/tty03
```

2335 Another example of "<>" is one that opens **/dev/tty** on file descriptor 3 for reading and writing:

```
2336     exec 3<> /dev/tty
```

2337 An example of creating a lock file for a critical code region:

```
2338     set -C  
2339     until      2> /dev/null > lockfile  
2340     do         sleep 30  
2341     done  
2342     set +C  
2343     perform critical function  
2344     rm lockfile
```

2345 Since **/dev/null** is not a regular file, no error is generated by redirecting to it in *noclobber* mode.

2346 Tilde expansion is not performed on a here-document because the data is treated as if it were
2347 enclosed in double quotes.

2.8 Exit Status and Errors

2.8.1 Consequences of Shell Errors

For a non-interactive shell, an error condition encountered by a special built-in (see Section 2.14 on page 96) or other type of utility shall cause the shell to write a diagnostic message to standard error and exit as shown in the following table:

Error	Special Built-In	Other Utilities
Shell language syntax error	Exits	Exits
Utility syntax error (option or operand error)	Exits	Does not exit
Redirection error	Exits	Does not exit
Variable assignment error	Exits	Does not exit
Expansion error	Exits	Exits
Command not found	N/A	May exit
Dot script not found	Exits	N/A

An expansion error is one that occurs when the shell expansions defined in Section 2.6 on page 49 are carried out (for example, "`${x!y}`", because '`!`' is not a valid operator); an implementation may treat these as syntax errors if it is able to detect them during tokenization, rather than during expansion.

If any of the errors shown as “shall exit” or “(may) exit” occur in a subshell, the subshell shall (or may exit) with a non-zero status, but the script containing the subshell shall not exit because of the error.

In all of the cases shown in the table, an interactive shell shall write a diagnostic message to standard error without exiting.

2.8.2 Exit Status for Commands

Each command has an exit status that can influence the behavior of other shell commands. The exit status of commands that are not utilities is documented in this section. The exit status of the standard utilities is documented in their respective sections.

If a command is not found, the exit status shall be 127. If the command name is found, but it is not an executable utility, the exit status shall be 126. Applications that invoke utilities without using the shell should use these exit status values to report similar errors.

If a command fails during word expansion or redirection, its exit status shall be greater than zero.

Internally, for purposes of deciding whether a command exits with a non-zero exit status, the shell shall recognize the entire status value retrieved for the command by the equivalent of the `wait()` function `WEXITSTATUS` macro (as defined in the System Interfaces volume of IEEE Std. 1003.1-200x). When reporting the exit status with the special parameter '`?`', the shell shall report the full eight bits of exit status available. The exit status of a command that terminated because it received a signal shall be reported as greater than 128.

Rationale

There is a historical difference in *sh* and *ksh* non-interactive error behavior. When a command named in a script is not found, some implementations of *sh* exit immediately, but *ksh* continues with the next command. Thus, this volume of IEEE Std. 1003.1-200x says that the shell “may” exit in this case. This puts a small burden on the programmer, who has to test for successful completion following a command if it is important that the next command not be executed if the previous command was not found. If it is important for the command to have been found, it was probably also important for it to complete successfully. The test for successful completion would not need to change.

Historically, shells have returned an exit status of $128+n$, where n represents the signal number. Since signal numbers are not standardized, there is no portable way to determine which signal caused the termination. Also, it is possible for a command to exit with a status in the same range of numbers that the shell would use to report that the command was terminated by a signal. Implementations are encouraged to choose exit values greater than 256 to indicate programs that terminate by a signal so that the exit status cannot be confused with an exit status generated by a normal termination.

Historical shells make the distinction between “utility not found” and “utility found but cannot execute” in their error messages. By specifying two seldomly used exit status values for these cases, 127 and 126 respectively, this gives an application the opportunity to make use of this distinction without having to parse an error message that would probably change from locale to locale. The *command*, *env*, *nohup*, and *xargs* utilities in this volume of IEEE Std. 1003.1-200x have also been specified to use this convention.

When a command fails during word expansion or redirection, most historical implementations exit with a status of 1. However, there was some sentiment that this value should probably be much higher so that an application could distinguish this case from the more normal exit status values. Thus, the language “greater than zero” was selected to allow either method to be implemented.

2.9 Shell Commands

This section describes the basic structure of shell commands. The following command descriptions each describe a format of the command that is only used to aid the reader in recognizing the command type, and does not formally represent the syntax. Each description discusses the semantics of the command; for a formal definition of the command language, consult Section 2.10 on page 82.

A *command* is one of the following:

- *Simple command* (see Section 2.9.1)
- *Pipeline* (see Section 2.9.2 on page 72)
- *List or compound-list* (see Section 2.9.3 on page 73)
- *Compound command* (see Section 2.9.4 on page 75)
- *Function definition* (see Section 2.9.5 on page 79)

Unless otherwise stated, the exit status of a command is that of the last simple command executed by the command. There is no limit on the size of any shell command other than that imposed by the underlying system (memory constraints, {ARG_MAX}, and so on).

Rationale

A description of an “empty command” was removed from an early proposal because it is only relevant in the cases of *sh -c " "*, *system(" ")*, or an empty shell-script file (such as the implementation of *true* on some historical systems). Since it is no longer mentioned in this volume of IEEE Std. 1003.1-200x, it falls into the silently unspecified category of behavior where implementations can continue to operate as they have historically, but conforming applications do not construct empty commands. (However, note that *sh* does explicitly state an exit status for an empty string or file.) In an interactive session or a script with other commands, extra <newline>s or semicolons, such as;

```
$ false
$
$ echo $?
1
```

would not qualify as the empty command described here because they would be consumed by other parts of the grammar.

2.9.1 Simple Commands

A *simple command* is a sequence of optional variable assignments and redirections, in any sequence, optionally followed by words and redirections, terminated by a control operator.

When a given simple command is required to be executed (that is, when any conditional construct such as an AND-OR list or a **case** statement has not bypassed the simple command), the following expansions, assignments, and redirections are all performed from the beginning of the command text to the end:

1. The words that are recognized as variable assignments or redirections according to Section 2.10.2 on page 82 are saved for processing in steps 3 and 4.
2. The words that are not variable assignments or redirections shall be expanded. If any fields remain following their expansion, the first field shall be considered the command name and remaining fields are the arguments for the command.

3. Redirections shall be performed as described in Section 2.7 on page 60.

4. Each variable assignment shall be expanded for tilde expansion, parameter expansion, command substitution, arithmetic expansion, and quote removal prior to assigning the value.

In the preceding list, the order of steps 3 and 4 may be reversed for the processing of special built-in utilities; see Section 2.14 on page 96.

If no command name results, variable assignments shall affect the current execution environment. Otherwise, the variable assignments shall be exported for the execution environment of the command and shall not affect the current execution environment (except for special built-ins). If any of the variable assignments attempt to assign a value to a read-only variable, a variable assignment error occurs. See Section 2.8.1 on page 65 for the consequences of these errors.

If there is no command name, any redirections shall be performed in a subshell environment; it is unspecified whether this subshell environment is the same one as that used for a command substitution within the command. (To affect the current execution environment, see the *exec* on page 107 special built-in.) If any of the redirections performed in the current shell execution environment fail, the command shall immediately fail with an exit status greater than zero, and the shell shall write an error message indicating the failure. See Section 2.8.1 on page 65 for the consequences of these failures on interactive and non-interactive shells.

If there is a command name, execution shall continue as described in Section 2.9.1.1 on page 69. If there is no command name, but the command contained a command substitution, the command shall complete with the exit status of the last command substitution performed. Otherwise, the command shall complete with a zero exit status.

Rationale

The enumerated list is used only when the command is actually going to be executed. For example, in:

```
true || $foo *
```

no expansions are performed.

The following example illustrates both how a variable assignment without a command name affects the current execution environment, and how an assignment with a command name only affects the execution environment of the command:

```
$ x=red
$ echo $x
red
$ export x
$ sh -c 'echo $x'
red
$ x=blue sh -c 'echo $x'
blue
$ echo $x
red
```

This next example illustrates that redirections without a command name are still performed:


```

2496      $ ls foo
2497      ls: foo: no such file or directory
2498      $ > foo
2499      $ ls foo
2500      foo

```

2501 A command without a command name, but one that includes a command substitution, has an
 2502 exit status of the last command substitution that the shell performed. For example:

```

2503      if      x=$(command)
2504      then    ...
2505      fi

```

2506 An example of redirections without a command name being performed in a subshell shows that
 2507 the here-document does not disrupt the standard input of the **while** loop:

```

2508      IFS=:
2509      while   read a b
2510      do      echo $a
2511              <<-eof
2512              Hello
2513              eof
2514      done </etc/passwd

```

2515 Some examples of commands without command names in AND-OR lists:

```

2516      > foo || {
2517          echo "error: foo cannot be created" >&2
2518          exit 1
2519      }
2520      # set saved if /vmunix.save exists
2521      test -f /vmunix.save && saved=1

```

2522 Command substitution and redirections without command names both occur in subshells, but
 2523 they are not necessarily the same ones. For example, in:

```

2524      exec 3> file
2525      var=$(echo foo >&3) 3>&1

```

2526 it is unspecified whether **foo** is echoed to the file or to standard output.

2527 2.9.1.1 Command Search and Execution

2528 If a simple command results in a command name and an optional list of arguments, the
 2529 following actions shall be performed:

- 2530 1. If the command name does not contain any slashes, the first successful step in the
 2531 following sequence shall occur:
 - 2532 a. If the command name matches the name of a special built-in utility, that special
 2533 built-in utility shall be invoked.
 - 2534 b. If the command name matches the name of a function known to this shell, the
 2535 function shall be invoked as described in Section 2.9.5 on page 79. If the
 2536 implementation has provided a standard utility in the form of a function, it shall not
 2537 be recognized at this point. It shall be invoked in conjunction with the path search in
 2538 step 1d.

- c. If the command name matches the name of a utility listed in the following table, that utility shall be invoked.

<i>alias</i>	<i>false</i>	<i>jobs</i>	<i>true</i>
<i>bg</i>	<i>fc</i>	<i>kill</i>	<i>umask</i>
<i>cd</i>	<i>fg</i>	<i>newgrp</i>	<i>unalias</i>
<i>command</i>	<i>getopts</i>	<i>read</i>	<i>wait</i>

- d. Otherwise, the command is searched for using the *PATH* environment variable as described in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 8, Environment Variables:

- i. If the search is successful:

- a. If the system has implemented the utility as a regular built-in or as a shell function, it shall be invoked at this point in the path search.
- b. Otherwise, the shell executes the utility in a separate utility environment (see Section 2.12 on page 90) with actions equivalent to calling the *execve()* function as defined in the System Interfaces volume of IEEE Std. 1003.1-200x with the *path* argument set to the path name resulting from the search, *arg0* set to the command name, and the remaining arguments set to the operands, if any.

If the *execve()* function fails due to an error equivalent to the [ENOEXEC] error defined in the System Interfaces volume of IEEE Std. 1003.1-200x, the shell shall execute a command equivalent to having a shell invoked with the command name as its first operand, along with any remaining arguments passed along. If the executable file is not a text file, the shell may bypass this command execution, write an error message, and return an exit status of 126.

Once a utility has been searched for and found (either as a result of this specific search or as part of an unspecified shell start-up activity), an implementation may remember its location and need not search for the utility again unless the *PATH* variable has been the subject of an assignment. If the remembered location fails for a subsequent invocation, the shell shall repeat the search to find the new location for the utility, if any.

- ii. If the search is unsuccessful, the command shall fail with an exit status of 127 and the shell shall write an error message.

2. If the command name contains at least one slash, the shell shall execute the utility in a separate utility environment with actions equivalent to calling the *execve()* function defined in the System Interfaces volume of IEEE Std. 1003.1-200x with the *path* and *arg0* arguments set to the command name, and the remaining arguments set to the operands, if any.

If the *execve()* function fails due to an error equivalent to the [ENOEXEC] error, the shell shall execute a command equivalent to having a shell invoked with the command name as its first operand, along with any remaining arguments passed along. If the executable file is not a text file, the shell may bypass this command execution, write an error message, and return an exit status of 126.

Rationale

This description requires that the shell can execute shell scripts directly, even if the underlying system does not support the common "#!" interpreter convention. That is, if file **foo** contains shell commands and is executable, the following executes **foo**:

```
./foo
```

The command search shown here does not match all historical implementations. A more typical sequence has been:

- Any built-in (special or regular)
- Functions
- Path search for executable files

But there are problems with this sequence. Since the programmer has no idea in advance which utilities might have been built into the shell, a function cannot be used to override portably a utility of the same name. (For example, a function named *cd* cannot be written for many historical systems.) Furthermore, the *PATH* variable is partially ineffective in this case, and only a path name with a slash can be used to ensure a specific executable file is invoked.

After the *execve()* failure described, the shell normally executes the file as a shell script. Some implementations, however, attempt to detect whether the file is actually a script and not an executable from some other architecture. The method used by the KornShell is allowed by the text that indicates non-text files may be bypassed.

The sequence selected for this volume of IEEE Std. 1003.1-200x acknowledges that special built-ins cannot be overridden, but gives the programmer full control over which versions of other utilities are executed. It provides a means of suppressing function lookup (via the *command* utility) for the user's own functions and ensures that any regular built-ins or functions provided by the implementation are under the control of the path search. The mechanisms for associating built-ins or functions with executable files in the path are not specified by this volume of IEEE Std. 1003.1-200x, but the wording requires that if either is implemented, the application is not able to distinguish a function or built-in from an executable (other than in terms of performance, presumably). The implementation ensures that all effects specified by this volume of IEEE Std. 1003.1-200x resulting from the invocation of the regular built-in or function (interaction with the environment, variables, traps, and so on) are identical to those resulting from the invocation of an executable file.

Examples

Consider three versions of the *ls* utility:

1. The application includes a shell function named *ls*.
2. The user writes a utility named *ls* and puts it in **/fred/bin**.
3. The example implementation provides *ls* as a regular shell built-in that is invoked (either by the shell or directly by *exec*) when the path search reaches the directory **/posix/bin**.

If *PATH*=**/posix/bin**, various invocations yield different versions of *ls*:

2620
2621
2622
2623
2624
2625
2626

Invocation	Version of <i>ls</i>
<i>ls</i> (from within application script)	(1) function
<i>command ls</i> (from within application script)	(3) built-in
<i>ls</i> (from within makefile called by application)	(3) built-in
<i>system("ls")</i>	(3) built-in
<i>PATH="/fred/bin:\$PATH" ls</i>	(2) user's version

2627 2.9.2 Pipelines

2628 A *pipeline* is a sequence of one or more commands separated by the control operator ' | '. The
2629 standard output of all but the last command shall be connected to the standard input of the next
2630 command.

2631 The format for a pipeline is:

2632 [!] *command1* [| *command2* ...]

2633 The standard output of *command1* shall be connected to the standard input of *command2*. The
2634 standard input, standard output, or both of a command shall be considered to be assigned by the
2635 pipeline before any redirection specified by redirection operators that are part of the command
2636 (see Section 2.7 on page 60).

2637 If the pipeline is not in the background (see Section 2.9.3.1 on page 74), the shell shall wait for the
2638 last command specified in the pipeline to complete, and may also wait for all commands to
2639 complete.

2640 Exit Status

2641 If the reserved word ! does not precede the pipeline, the exit status shall be the exit status of the
2642 last command specified in the pipeline. Otherwise, the exit status shall be the logical NOT of the
2643 exit status of the last command. That is, if the last command returns zero, the exit status shall be
2644 1; if the last command returns greater than zero, the exit status shall be zero.

2645 Rationale

2646 Because pipeline assignment of standard input or standard output or both takes place before
2647 redirection, it can be modified by redirection. For example:

2648 \$ *command1* 2>&1 | *command2*

2649 sends both the standard output and standard error of *command1* to the standard input of
2650 *command2*.

2651 The reserved word ! allows more flexible testing using AND and OR lists.

2652 It was suggested that it would be better to return a non-zero value if any command in the
2653 pipeline terminates with non-zero status (perhaps the bitwise-inclusive OR of all return values).
2654 However, the choice of the last-specified command semantics are historical practice and would
2655 cause applications to break if changed. An example of historical behavior:

2656 \$ sleep 5 | (exit 4)
2657 \$ echo \$?
2658 4
2659 \$ (exit 4) | sleep 5
2660 \$ echo \$?
2661 0

2662 **2.9.3 Lists**

2663 An *AND-OR list* is a sequence of one or more pipelines separated by the operators "&&" and
 2664 "||".

2665 A *list* is a sequence of one or more AND-OR lists separated by the operators ';' and '&' and
 2666 optionally terminated by ';', '&', or <newline>.

2667 The operators "&&" and "||" shall have equal precedence and are evaluated from beginning to
 2668 end. For example, both of the following commands write solely **bar** to standard output:

```
2669     false && echo foo || echo bar
2670     true  || echo foo && echo bar
```

2671 A ';' or <newline> character terminator shall cause the preceding AND-OR list to be executed
 2672 sequentially; an '&' shall cause asynchronous execution of the preceding AND-OR list.

2673 The term *compound-list* is derived from the grammar in Section 2.10 on page 82; it is equivalent to
 2674 a sequence of *lists*, separated by <newline> characters, that can be preceded or followed by an
 2675 arbitrary number of <newline> characters.

2676 **Examples**

2677 The following is an example that illustrates <newline> characters in compound-lists:

```
2678     while
2679         # a couple of <newline>s
2680
2681         # a list
2682         date && who || ls; cat file
2683         # a couple of <newline>s
2684
2685         # another list
2686         wc file > output & true
2687
2688     do
2689         # 2 lists
2690         ls
2691         cat file
2692     done
```

2690 **Rationale**

2691 The equal precedence of "&&" and "||" is historical practice. The standard developers
 2692 evaluated the model used more frequently in high-level programming languages, such as C, to
 2693 allow the shell logical operators to be used for complex expressions in an unambiguous way, but
 2694 they could not allow historical scripts to break in the subtle way unequal precedence might
 2695 cause. Some arguments were posed concerning the "{" or "(" groupings that are required
 2696 historically. There are some disadvantages to these groupings:

- 2697 • The "(" can be expensive, as they spawn other processes on some systems. This
 2698 performance concern is primarily an implementation issue.
- 2699 • The "{" braces are not operators (they are reserved words) and require a trailing space after
 2700 each '{', and a semicolon before each '}'. Most programmers (and certainly interactive
 2701 users) have avoided braces as grouping constructs because of the problematic syntax
 2702 required. Braces were not changed to operators because that would generate compatibility
 2703 issues even greater than the precedence question; braces appear outside the context of a
 2704 keyword in many shell scripts.

2705 2.9.3.1 *Asynchronous Lists*

2706 If a command is terminated by the control operator ampersand ('&'), the shell shall execute the
 2707 command asynchronously in a subshell. This means that the shell shall not wait for the
 2708 command to finish before executing the next command.

2709 The format for running a command in the background is:

2710 `command1 & [command2 & ...]`

2711 The standard input for an asynchronous list, before any explicit redirections are performed, shall
 2712 be considered to be assigned to a file that has the same properties as **/dev/null**. If it is an
 2713 interactive shell, this need not happen. In all cases, explicit redirection of standard input shall
 2714 override this activity.

2715 When an element of an asynchronous list (the portion of the list ended by an ampersand, such as
 2716 *command1*, above) is started by the shell, the process ID of the last command in the asynchronous
 2717 list element shall become known in the current shell execution environment; see Section 2.12 on
 2718 page 90. This process ID shall remain known until:

- 2719 1. The command terminates and the application waits for the process ID.
- 2720 2. Another asynchronous list invoked before "\$!" (corresponding to the previous
 2721 asynchronous list) is expanded in the current execution environment.

2722 The implementation need not retain more than the {CHILD_MAX} most recent entries in its list
 2723 of known process IDs in the current shell execution environment.

2724 **Exit Status**

2725 The exit status of an asynchronous list shall be zero.

2726 **Rationale**

2727 The grammar treats a construct such as:

2728 `foo & bar & bam &`

2729 as one “asynchronous list”, but since the status of each element is tracked by the shell, the term
 2730 “element of an asynchronous list” was introduced to identify just one of the **foo**, **bar**, or **bam**
 2731 portions of the overall list.

2732 Unless the implementation has an internal limit, such as {CHILD_MAX}, on the retained process
 2733 IDs, it would require unbounded memory for the following example:

```
2734 while true
2735 do     foo & echo $!
2736 done
```

2737 The treatment of the signals SIGINT and SIGQUIT with asynchronous lists is described in
 2738 Section 2.11 on page 89.

2739 Since the connection of the input to the equivalent of **/dev/null** is considered to occur before
 2740 redirections, the following script would produce no output:

```
2741 exec < /etc/passwd
2742 cat <&0 &
2743 wait
```

2744 2.9.3.2 *Sequential Lists*

2745 Commands that are separated by a semicolon (;) shall be executed sequentially.

2746 The format for executing commands sequentially shall be:

2747 `command1 [; command2] ...`

2748 Each command shall be expanded and executed in the order specified.

2749 **Exit Status**

2750 The exit status of a sequential list shall be the exit status of the last command in the list.

2751 2.9.3.3 *AND Lists*

2752 The control operator " && " denotes an AND list. The format shall be:

2753 `command1 [&& command2] ...`

2754 First *command1* shall be executed. If its exit status is zero, *command2* shall be executed, and so on,
 2755 until a command has a non-zero exit status or there are no more commands left to execute. The
 2756 commands are expanded only if they are executed.

2757 **Exit Status**

2758 The exit status of an AND list shall be the exit status of the last command that is executed in the
 2759 list.

2760 2.9.3.4 *OR Lists*

2761 The control operator " || " denotes an OR List. The format shall be:

2762 `command1 [|| command2] ...`

2763 First, *command1* shall be executed. If its exit status is non-zero, *command2* shall be executed, and
 2764 so on, until a command has a zero exit status or there are no more commands left to execute.

2765 **Exit Status**

2766 The exit status of an OR list shall be the exit status of the last command that is executed in the
 2767 list.

2768 2.9.4 **Compound Commands**

2769 The shell has several programming constructs that are *compound commands*, which provide
 2770 control flow for commands. Each of these compound commands has a reserved word or control
 2771 operator at the beginning, and a corresponding terminator reserved word or operator at the end.
 2772 In addition, each can be followed by redirections on the same line as the terminator. Each
 2773 redirection shall apply to all the commands within the compound command that do not
 2774 explicitly override that redirection.

2775 2.9.4.1 *Grouping Commands*

2776 The format for grouping commands is as follows:

2777 (*compound-list*) Execute *compound-list* in a subshell environment; see Section 2.12 on page
 2778 90. Variable assignments and built-in commands that affect the
 2779 environment shall not remain in effect after the list finishes.

2780 { *compound-list*;} Execute *compound-list* in the current process environment. The semicolon
 2781 shown here is an example of a control operator delimiting the } reserved
 2782 word. Other delimiters are possible, as shown in Section 2.10 on page 82;
 2783 a <newline> character is frequently used.

2784 **Exit Status**

2785 The exit status of a grouping command shall be the exit status of *list*.

2786 **Rationale**

2787 The semicolon shown {*compound-list*;} is an example of a control operator delimiting the }
 2788 reserved word. Other delimiters are possible, as shown in Section 2.10 on page 82; <newline> is
 2789 frequently used.

2790 A proposal was made to use the <do-done> construct in all cases where command grouping in
 2791 the current process environment is performed, identifying it as a construct for the grouping
 2792 commands, as well as for shell functions. This was not included because the shell already has a
 2793 grouping construct for this purpose ("{}"), and changing it would have been counter-
 2794 productive.

2795 2.9.4.2 *For Loop*

2796 The **for** loop executes a sequence of commands for each member in a list of *items*. The **for** loop
 2797 requires that the reserved words **do** and **done** be used to delimit the sequence of commands.

2798 The format for the **for** loop is as follows:

```
2799     for name [ in word ... ]
2800     do
2801         compound-list
2802     done
```

2803 First, the list of words following **in** shall be expanded to generate a list of items. Then, the
 2804 variable *name* shall be set to each item, in turn, and the *compound-list* executed each time. If no
 2805 items result from the expansion, the *compound-list* shall not be executed. Omitting:

```
2806     in word...
```

2807 is equivalent to:

```
2808     in "$@"
```

2809 **Exit Status**

2810 The exit status of a **for** command shall be the exit status of the last command that executes. If
 2811 there are no items, the exit status shall be zero.

Rationale

The format is shown with generous usage of <newline>s. See the grammar in Section 2.10 on page 82 for a precise description of where <newline>s and semicolons can be interchanged.

Some historical implementations support '**{**' and '**}**' as substitutes for **do** and **done**. The standard developers chose to omit them, even as an obsolescent feature. (Note that these substitutes were only for the **for** command; the **while** and **until** commands could not use them historically because they are followed by compound-lists that may contain "{ ... }" grouping commands themselves.)

The reserved word pair **do ... done** was selected rather than **do ... od** (which would have matched the spirit of **if ... fi** and **case ... esac**) because *od* is already the name of a standard utility.

2.9.4.3 Case Conditional Construct

The conditional construct **case** shall execute the *compound-list* corresponding to the first one of several *patterns* (see Section 2.13 on page 92) that is matched by the string resulting from the tilde expansion, parameter expansion, command substitution, arithmetic expansion, and quote removal of the given word. The reserved word **in** shall denote the beginning of the patterns to be matched. Multiple patterns with the same *compound-list* shall be delimited by the '|' symbol. The control operator ')' terminates a list of patterns corresponding to a given action. The *compound-list* for each list of patterns, with the possible exception of the last, shall be terminated with ";". The **case** construct terminates with the reserved word **esac** (**case** reversed).

The format for the **case** construct is as follows:

```
case word in
    [(]pattern[) compound-list;;
    [[(]pattern[ | pattern] ... ) compound-list;;] ...
    [[(]pattern[ | pattern] ... ) compound-list]
esac
```

The ";" is optional for the last *compound-list*.

In order from the beginning to the end of the **case** statement, each *pattern* that labels a *compound-list* shall be subjected to tilde expansion, parameter expansion, command substitution, and arithmetic expansion, and the result of these expansions shall be compared against the expansion of *word*, according to the rules described in Section 2.13 on page 92 (which also describes the effect of quoting parts of the pattern). After the first match, no more patterns shall be expanded, and the *compound-list* shall be executed. The order of expansion and comparison of multiple *patterns* that label a *compound-list* statement is unspecified.

Exit Status

The exit status of **case** shall be zero if no patterns are matched. Otherwise, the exit status shall be the exit status of the last command executed in the *compound-list*.

Rationale

An optional left parenthesis before *pattern* was added to allow numerous historical KornShell scripts to conform. At one time, using the leading parenthesis was required if the **case** statement was to be embedded within a "`$ ()`" command substitution; this is no longer the case with the POSIX shell. Nevertheless, many historical scripts use the left parenthesis, if only because it makes matching-parenthesis searching easier in *vi* and other editors. This is a relatively simple implementation change that is upward compatible for all scripts.

Consideration was given to requiring *break* inside the *compound-list* to prevent falling through to the next pattern action list. This was rejected as being nonexistent practice. An interesting undocumented feature of the KornShell is that using "`;&`" instead of "`;;`" as a terminator causes the exact opposite behavior—the flow of control continues with the next *compound-list*.

The pattern '`*`', given as the last pattern in a **case** construct, is equivalent to the default case in a C-language **switch** statement.

The grammar shows that reserved words can be used as patterns, even if one is the first word on a line. Obviously, the reserved word **esac** cannot be used in this manner.

2.9.4.4 If Conditional Construct

The **if** command shall execute a *compound-list* and use its exit status to determine whether to execute another *compound-list*.

The format for the **if** construct is as follows:

```
if compound-list
then
    compound-list
[elif compound-list
then
    compound-list] ...
[else
    compound-list]
```

The **if** *compound-list* shall be executed; if its exit status is zero, the **then** *compound-list* shall be executed and the command shall complete. Otherwise, each **elif** *compound-list* shall be executed, in turn, and if its exit status is zero, the **then** *compound-list* shall be executed and the command shall complete. Otherwise, the **else** *compound-list* shall be executed.

Exit Status

The exit status of the **if** command shall be the exit status of the **then** or **else** *compound-list* that was executed, or zero, if none was executed.

Rationale

The precise format for the command syntax is described in Section 2.10 on page 82.

2.9.4.5 While Loop

The **while** loop shall continuously execute one *compound-list* as long as another *compound-list* has a zero exit status.

The format of the **while** loop is as follows:

```

2889         while compound-list-1
2890         do
2891             compound-list-2
2892         done

```

2893 The *compound-list-1* shall be executed, and if it has a non-zero exit status, the **while** command
 2894 shall complete. Otherwise, the *compound-list-2* shall be executed, and the process shall repeat.

2895 **Exit Status**

2896 The exit status of the **while** loop shall be the exit status of the last *compound-list-2* executed, or
 2897 zero if none was executed.

2898 **Rationale**

2899 The precise format for the command syntax is described in Section 2.10 on page 82.

2900 **2.9.4.6 Until Loop**

2901 The **until** loop shall continuously execute one *compound-list* as long as another *compound-list* has
 2902 a non-zero exit status.

2903 The format of the **until** loop is as follows:

```

2904         until compound-list-1
2905         do
2906             compound-list-2
2907         done

```

2908 The *compound-list-1* shall be executed, and if it has a zero exit status, the **until** command
 2909 completes. Otherwise, the *compound-list-2* shall be executed, and the process repeats.

2910 **Exit Status**

2911 The exit status of the **until** loop shall be the exit status of the last *compound-list-2* executed, or
 2912 zero if none was executed.

2913 **Rationale**

2914 The precise format for the command syntax is described in Section 2.10 on page 82.

2915 **2.9.5 Function Definition Command**

2916 A function is a user-defined name that is used as a simple command to call a compound
 2917 command with new positional parameters. A function is defined with a *function definition*
 2918 *command*.

2919 The format of a function definition command is as follows:

```

2920     fname() compound-command[io-redirect ...]

```

2921 The function is named *fname*; the application shall ensure that it is a name (see the System
 2922 Interface Definitions volume of IEEE Std. 1003.1-200x, Section 3.234, Name). An implementation
 2923 may allow other characters in a function name as an extension. The implementation shall
 2924 maintain separate name spaces for functions and variables.

2925 The argument *compound-command* represents a compound command, as described in Section
 2926 2.9.4 on page 75.

When the function is declared, none of the expansions in Section 2.6 on page 49 shall be performed on the text in *compound-command* or *io-redirect*; all expansions shall be performed as normal each time the function is called. Similarly, the optional *io-redirect* redirections and any variable assignments within *compound-command* shall be performed during the execution of the function itself, not the function definition. See Section 2.8.1 on page 65 for the consequences of failures of these operations on interactive and non-interactive shells.

When a function is executed, it shall have the syntax-error and variable-assignment properties described for special built-in utilities in the enumerated list at the beginning of Section 2.14 on page 96.

The *compound-command* shall be executed whenever the function name is specified as the name of a simple command (see Section 2.9.1.1 on page 69). The operands to the command temporarily shall become the positional parameters during the execution of the *compound-command*; the special parameter '#' also shall be changed to reflect the number of operands. The special parameter 0 shall be unchanged. When the function completes, the values of the positional parameters and the special parameter '#' shall be restored to the values they had before the function was executed. If the special built-in *return* is executed in the *compound-command*, the function completes and execution shall resume with the next command after the function call.

Exit Status

The exit status of a function definition shall be zero if the function was declared successfully; otherwise, it shall be greater than zero. The exit status of a function invocation shall be the exit status of the last command executed by the function.

Rationale

The description of functions in an early proposal was based on the notion that functions should behave like miniature shell scripts; that is, except for sharing variables, most elements of an execution environment should behave as if they were a new execution environment, and changes to these should be local to the function. For example, traps and options should be reset on entry to the function, and any changes to them do not affect the traps or options of the caller. There were numerous objections to this basic idea, and the opponents asserted that functions were intended to be a convenient mechanism for grouping common commands that were to be executed in the current execution environment, similar to the execution of the *dot* special built-in.

It was also pointed out that the functions described in that early proposal did not provide a local scope for everything a new shell script would, such as the current working directory, or *umask*, but instead provided a local scope for only a few select properties. The basic argument was that if a local scope is needed for the execution environment, the mechanism already existed: the application can put the commands in a new shell script and call that script. All historical shells that implemented functions, other than the KornShell, have implemented functions that operate in the current execution environment. Because of this, traps and options have a global scope within a shell script. Local variables within a function were considered and included in another early proposal (controlled by the special built-in *local*), but were removed because they do not fit the simple model developed for functions and because there was some opposition to adding yet another new special built-in that was not part of historical practice. Implementations should reserve the identifier *local* (as well as *typeset*, as used in the KornShell) in case this local variable mechanism is adopted in a future version of this volume of IEEE Std. 1003.1-200x.

A separate issue from the execution environment of a function is the availability of that function to child shells. A few objectors maintained that just as a variable can be shared with child shells

by exporting it, so should a function—and so this capability has been added to this volume of IEEE Std. 1003.1-200x. In early proposals, the *export* command therefore had a *-f* flag for exporting functions. Functions that were exported were to be put into the environment as *name()=value* pairs, and upon invocation, the shell would scan the environment for these and automatically define these functions. This facility was strongly opposed and was omitted. Some of the arguments against exportable functions were as follows:

- There was little historical practice. The Ninth Edition shell provided them, but there was controversy over how well it worked.
- There are numerous security problems associated with functions appearing in the environment of a user and overriding standard utilities or the utilities owned by the application.
- There was controversy over requiring *make* to import functions, where it has historically used an *exec* function for many of its command line executions.
- Functions can be big and the environment is of a limited size. (The counter-argument was that functions are no different than variables in terms of size: there can be big ones, and there can be small ones—and just as one does not export huge variables, one does not export huge functions. However, this might not apply to the average shell-function writer, who typically writes much larger functions than variables.)

As far as can be determined, the functions in this volume of IEEE Std. 1003.1-200x match those in System V. Earlier versions of the KornShell had two methods of defining functions:

```
function fname { compound-list }
```

and:

```
fname() { compound-list }
```

The latter used the same definition as this volume of IEEE Std. 1003.1-200x, but differed in semantics, as described previously. The current edition of the KornShell aligns the latter syntax with this volume of IEEE Std. 1003.1-200x and keeps the former as is.

The name space for functions is limited to that of a *name* because of historical practice. Complications in defining the syntactic rules for the function definition command and in dealing with known extensions such as the "*@()*" usage in the KornShell prevented the name space from being widened to a *word*. Using functions to support synonyms such as the "*!!*" and '*%*' usage in the C shell is thus disallowed to portable applications, but acceptable as an extension. For interactive users, the aliasing facilities in this volume of IEEE Std. 1003.1-200x should be adequate for this purpose. It is recognized that the name space for utilities in the file system is wider than that currently supported for functions, if the portable file name character set guidelines are ignored, but it did not seem useful to mandate extensions in systems for so little benefit to portable applications.

The "*()*" in the function definition command consists of two operators. Therefore, intermixing <blank>s with the *fname*, '*(*', and '*)*' is allowed, but unnecessary.

An example of how a function definition can be used wherever a simple command is allowed:

```
# If variable i is equal to "yes",
# define function foo to be ls -l
#
[ "$i" = yes ] && foo() {
    ls -l
}
```

3019 2.10 Shell Grammar

3020 The following grammar defines the Shell Command Language. This formal syntax shall take
3021 precedence over the preceding text syntax description.

3022 2.10.1 Shell Grammar Lexical Conventions

3023 The input language to the shell must be first recognized at the character level. The resulting
3024 tokens shall be classified by their immediate context according to the following rules (applied in
3025 order). These rules are used to determine what a “token” is that is subject to parsing at the
3026 token level. The rules for token recognition in Section 2.3 on page 39 shall apply.

- 3027 1. A <newline> character shall be returned as the token identifier **NEWLINE**.
- 3028 2. If the token is an operator, the token identifier for that operator shall result.
- 3029 3. If the string consists solely of digits and the delimiter character is one of '<' or '>', the
3030 token identifier **IO_NUMBER** shall be returned.
- 3031 4. Otherwise, the token identifier **TOKEN** results.

3032 Further distinction on **TOKEN** is context-dependent. It may be that the same **TOKEN** yields
3033 **WORD**, a **NAME**, an **ASSIGNMENT**, or one of the reserved words below, dependent upon the
3034 context. Some of the productions in the grammar below are annotated with a rule number from
3035 the following list. When a **TOKEN** is seen where one of those annotated productions could be
3036 used to reduce the symbol, the applicable rule shall be applied to convert the token identifier
3037 type of the **TOKEN** to a token identifier acceptable at that point in the grammar. The reduction
3038 shall then proceed based upon the token identifier type yielded by the rule applied. When more
3039 than one rule applies, the highest numbered rule shall apply (which in turn may refer to another
3040 rule). (Note that except in rule 7, the presence of an '=' in the token has no effect.)

3041 The **WORD** tokens shall have the word expansion rules applied to them immediately before the
3042 associated command is executed, not at the time the command is parsed.

3043 2.10.2 Shell Grammar Rules

- 3044 1. [Command Name]

3045 When the **TOKEN** is exactly a reserved word, the token identifier for that reserved word
3046 shall result. Otherwise, the token **WORD** shall be returned. Also, if the parser is in any
3047 state where only a reserved word could be the next correct token, proceed as above. This
3048 rule applies rather narrowly: when a compound list is terminated by some clear delimiter
3049 (such as the closing **fi** of an inner **if_clause**) then it would apply; where the compound list
3050 might continue (as in after a ';'), rule 7a (and consequently the first sentence of this rule)
3051 would apply. In many instances the two conditions are identical, but this part of this rule
3052 does not give license to treating a **WORD** as a reserved word unless it is in a place where a
3053 reserved word shall appear.

3054 **Note:** Because at this point quote marks are retained in the token, quoted strings
3055 cannot be recognized as reserved words. This rule also implies that reserved
3056 words are not recognized except in certain positions in the input, such as after a
3057 <newline> character or semicolon; the grammar presumes that if the reserved
3058 word is intended, it is properly delimited by the user, and does not attempt to
3059 reflect that requirement directly. Also note that line joining is done before
3060 tokenization, as described in Section 2.2.1 on page 36, so escaped <newline>s
3061 are already removed at this point.

- 3062 Rule 1 is not directly referenced in the grammar, but is referred to by other rules, or applies
 3063 globally.
- 3064 2. [Redirection to or from file name]
- 3065 The expansions specified in Section 2.7 on page 60 shall occur. As specified there, exactly
 3066 one field can result (or the result is unspecified), and there are additional requirements on
 3067 path name expansion.
- 3068 3. [Redirection from here-document]
- 3069 Quote removal shall be applied to the word to determine the delimiter that is used to find
 3070 the end of the here-document that begins after the next <newline> character.
- 3071 4. [Case statement termination]
- 3072 When the **TOKEN** is exactly the reserved word **esac**, the token identifier for **esac** shall
 3073 result. Otherwise, the token **WORD** shall be returned.
- 3074 5. [NAME in **for**]
- 3075 When the **TOKEN** meets the requirements for a name (see the System Interface Definitions
 3076 volume of IEEE Std. 1003.1-200x, Section 3.234, Name), the token identifier **NAME** shall
 3077 result. Otherwise, the token **WORD** shall be returned.
- 3078 6. [Third word of **for** and **case**]
- 3079 When the **TOKEN** is exactly the reserved word **in**, the token identifier for **in** shall result.
 3080 Otherwise, the token **WORD** shall be returned. (As indicated in the grammar, a *linebreak*
 3081 precedes the token **in**. If <newline> characters are present at the indicated location, it is
 3082 the token after them that is treated in this fashion.)
- 3083 7. [Assignment preceding command name]
- 3084 a. [When the first word]
- 3085 If the **TOKEN** does not contain the character '=' , rule 1 is applied. Otherwise, 7b
 3086 shall be applied.
- 3087 b. [Not the first word]
- 3088 If the **TOKEN** contains the equal sign character:
- 3089 — If it begins with '=' , the token **WORD** shall be returned.
- 3090 — If all the characters preceding '=' form a valid name (see the System Interface
 3091 Definitions volume of IEEE Std. 1003.1-200x, Section 3.234, Name), the token
 3092 **ASSIGNMENT_WORD** shall be returned. (Quoted characters cannot participate
 3093 in forming a valid name.)
- 3094 — Otherwise, it is unspecified whether it is **ASSIGNMENT_WORD** or **WORD** that
 3095 is returned.
- 3096 Assignment to the **NAME** shall occur as specified in Section 2.9.1 on page 67.
- 3097 8. [NAME in function]
- 3098 When the **TOKEN** is exactly a reserved word, the token identifier for that reserved word
 3099 shall result. Otherwise, when the **TOKEN** meets the requirements for a name, the token
 3100 identifier **NAME** shall result. Otherwise, rule 7 applies.
- 3101 9. [Body of function]

```

3102         Word expansion and assignment shall never occur, even when required by the rules above,
3103         when this rule is being parsed. Each TOKEN that might either be expanded or have
3104         assignment applied to it shall instead be returned as a single WORD consisting only of
3105         characters that are exactly the token described in Section 2.3 on page 39.

3106         /* -----
3107         The grammar symbols
3108         ----- */

3109         %token  WORD
3110         %token  ASSIGNMENT_WORD
3111         %token  NAME
3112         %token  NEWLINE
3113         %token  IO_NUMBER

3114         /* The following are the operators mentioned above. */

3115         %token  AND_IF      OR_IF      DSEMI
3116         /*      '&&'      '|'      ';'      */

3117         %token  DLESS      DGREAT      LESSAND      GREATAND      LESSGREAT      DLESSDASH
3118         /*      '<<'      '>>'      '<&'      '>&'      '<>'      '<<-'      */

3119         %token  CLOBBER
3120         /*      '>|'      */

3121         /* The following are the reserved words. */

3122         %token  If      Then      Else      Elif      Fi      Do      Done
3123         /*      'if'      'then'      'else'      'elif'      'fi'      'do'      'done'      */

3124         %token  Case      Esac      While      Until      For
3125         /*      'case'      'esac'      'while'      'until'      'for'      */

3126         /* These are reserved words, not operator tokens, and are
3127            recognized when reserved words are recognized. */

3128         %token  Lbrace      Rbrace      Bang
3129         /*      '{'      '}'      '!'      */

3130         %token  In
3131         /*      'in'      */

3132         /* -----
3133         The Grammar
3134         ----- */

3135         %start  complete_command
3136         %%
3137         complete_command : list separator
3138                           | list
3139                           ;
3140         list             : list separator_op and_or
3141                           |
3142                           ;
3143         and_or           :
3144                           | and_or AND_IF linebreak pipeline
3145                           | and_or OR_IF linebreak pipeline
3146                           ;

```



```

3147     pipeline      :      pipe_sequence
3148                   | Bang pipe_sequence
3149                   ;
3150     pipe_sequence  :      command
3151                   | pipe_sequence '|' linebreak command
3152                   ;
3153     command        : simple_command
3154                   | compound_command
3155                   | compound_command redirect_list
3156                   | function_definition
3157                   ;
3158     compound_command : brace_group
3159                   | subshell
3160                   | for_clause
3161                   | case_clause
3162                   | if_clause
3163                   | while_clause
3164                   | until_clause
3165                   ;
3166     subshell       : '(' compound_list ')'
3167                   ;
3168     compound_list   :      term
3169                   | newline_list term
3170                   |      term separator
3171                   | newline_list term separator
3172                   ;
3173     term            : term separator and_or
3174                   |      and_or
3175                   ;
3176     for_clause      : For name linebreak do_group
3177                   | For name linebreak in wordlist sequential_sep do_group
3178                   ;
3179     name            : NAME /* Apply rule 5 */
3180                   ;
3181     in              : In /* Apply rule 6 */
3182                   ;
3183     wordlist        : wordlist WORD
3184                   |      WORD
3185                   ;
3186     case_clause     : Case WORD linebreak in linebreak case_list Esac
3187                   | Case WORD linebreak in linebreak case_list_ns Esac
3188                   | Case WORD linebreak in linebreak Esac
3189                   ;
3190     case_list_ns    : case_list case_item_ns
3191                   | case_item_ns
3192                   ;
3193     case_list       : case_list case_item
3194                   | case_item
3195                   ;
3196     case_item_ns    : pattern ')' linebreak linebreak
3197                   | pattern ')' compound_list linebreak
3198                   | '(' pattern ')' linebreak linebreak

```

```

3199         | '(' pattern ')' compound_list linebreak
3200         ;
3201     case_item      : pattern ')' linebreak DSEMI linebreak
3202         | pattern ')' compound_list linebreak
3203         | '(' pattern ')' linebreak linebreak
3204         | '(' pattern ')' compound_list linebreak
3205         ;
3206     pattern        : WORD /* Apply rule 4 */
3207         | pattern '|' WORD /* Do not apply rule (4) */
3208         ;
3209     if_clause       : If compound_list Then compound_list else_part Fi
3210         | If compound_list Then compound_list Fi
3211         ;
3212     else_part       : Elif compound_list Then else_part
3213         | Else compound_list
3214         ;
3215     while_clause    : While compound_list do_group
3216         ;
3217     until_clause    : Until compound_list do_group
3218         ;
3219     function_definition : fname '(' ')' linebreak function_body
3220         ;
3221     function_body   : compound_command /* Apply rule 9 */
3222         | compound_command redirect_list /* Apply rule 9 */
3223         ;
3224     fname           : NAME /* Apply rule 8 */
3225         ;
3226     brace_group     : Lbrace compound_list Rbrace
3227         ;
3228     do_group        : Do compound_list Done
3229         ;
3230     simple_command  : cmd_prefix cmd_word cmd_suffix
3231         | cmd_prefix cmd_word
3232         | cmd_prefix
3233         | cmd_name cmd_suffix
3234         | cmd_name
3235         ;
3236     cmd_name        : WORD /* Apply rule 7a */
3237         ;
3238     cmd_word        : WORD /* Apply rule 7b */
3239         ;
3240     cmd_prefix      : io_redirect
3241         | cmd_prefix io_redirect
3242         | ASSIGNMENT_WORD
3243         | cmd_prefix ASSIGNMENT_WORD
3244         ;
3245     cmd_suffix      : io_redirect
3246         | cmd_suffix io_redirect
3247         | WORD
3248         | cmd_suffix WORD
3249         ;
3250     redirect_list   : io_redirect

```

```

3251      | redirect_list io_redirect
3252      ;
3253      io_redirect      :      io_file
3254      | IO_NUMBER io_file
3255      |      io_here
3256      | IO_NUMBER io_here
3257      ;
3258      io_file          :  '<'      filename
3259      | LESSAND      filename
3260      | '>'      filename
3261      | GREATAND     filename
3262      | DGREAT      filename
3263      | LESSGREAT    filename
3264      | CLOBBER      filename
3265      ;
3266      filename         : WORD                      /* Apply rule 2 */
3267      ;
3268      io_here          : DLESS      here_end
3269      | DLESSDASH here_end
3270      ;
3271      here_end         : WORD                      /* Apply rule 3 */
3272      ;
3273      newline_list     :      NEWLINE
3274      | newline_list NEWLINE
3275      ;
3276      linebreak        : newline_list
3277      | /* empty */
3278      ;
3279      separator_op     : '&'
3280      | ';'
3281      ;
3282      separator        : separator_op linebreak
3283      | newline_list
3284      ;
3285      sequential_sep   : ';' linebreak
3286      | newline_list
3287      ;

```

Rationale

There are several subtle aspects of this grammar where conventional usage implies rules about the grammar that in fact are not true.

For *compound_list*, only the forms that end in a *separator* allow a reserved word to be recognized, so usually only a *separator* can be used where a compound list precedes a reserved word (such as **Then**, **Else**, **Do** and **Rbrace**). Explicitly requiring a separator would disallow such valid (if rare) statements as:

```
if (false) then (echo x) else (echo y) fi
```

See the Note under special grammar rule 1.

Concerning the third sentence of rule (1) (“Also, if the parser ...”):

- This sentence applies rather narrowly: when a compound list is terminated by some clear delimiter (such as the closing **fi** of an inner **if_clause**) then it would apply; where the compound list might continue (as in after a **;**), rule (7a) (and consequently the first sentence of rule (1)) would apply. In many instances the two conditions are identical, but this part of rule (1) does not give license to treating a **WORD** as a reserved word unless it is in a place where a reserved word has to appear.

- The statement is equivalent to requiring that when the LR(1) lookahead set contains exactly one reserved word, it must be recognized if it is present. (Here “LR(1)” refers to the theoretical concepts, not to any real parser generator.)

For example, in the construct below, and when the parser is at the point marked with **^**, the only next legal token is **then** (this follows directly from the grammar rules):

```
if if...fi then ... fi
      ^
```

At that point, the **then** must be recognized as a reserved word.

(Depending on the parser generator actually used, “extra” reserved words may be in some lookahead sets. It does not really matter if they are recognized, or even if any possible reserved word is recognized in that state, because if it is recognized and is not in the (theoretical) LR(1) lookahead set, an error is ultimately detected. In the example above, if some other reserved word (for example, **while**) is also recognized, an error occurs later.

This is approximately equivalent to saying that reserved words are recognized after other reserved words (because it is after a reserved word that this condition occurs), but avoids the “except for ...” list that would be required for *case*, *for*, and so on. (Reserved words are of course recognized anywhere a *simple_command* can appear, as well. Other rules take care of the special cases of non-recognition, such as rule (4) for case statements.)

Note that the body of here-documents are handled by token recognition (see Section 2.3 on page 39) and do not appear in the grammar directly. (However, the here-document I/O redirection operator is handled as part of the grammar.)

The start symbol of the grammar (**complete_command**) represents either input from the command line or a shell script. It is repeatedly applied by the interpreter to its input and represents a single “chunk” of that input as seen by the interpreter.

3328 2.11 Signals and Error Handling

3329 When a command is in an asynchronous list, the shell shall prevent SIGQUIT and SIGINT
3330 signals from the keyboard from interrupting the command. Otherwise, signals shall have the
3331 values inherited by the shell from its parent (see also the *trap* on page 127 special built-in).

3332 When a signal for which a trap has been set is received while the shell is waiting for the
3333 completion of a utility executing a foreground command, the trap associated with that signal
3334 shall not be executed until after the foreground command has completed. When the shell is
3335 waiting, by means of the *wait* utility, for asynchronous commands to complete, the reception of a
3336 signal for which a trap has been set shall cause the *wait* utility to return immediately with an exit
3337 status >128, immediately after which the trap associated with that signal shall be taken.

3338 If multiple signals are pending for the shell for which there are associated trap actions, the order
3339 of execution of trap actions is unspecified.

2.12 Shell Execution Environment

A shell execution environment consists of the following:

- Open files inherited upon invocation of the shell, plus open files controlled by *exec*
- Working directory as set by *cd*
- File creation mask set by *umask*
- Current traps set by *trap*
- Shell parameters that are set by variable assignment (see the *set* on page 117 special built-in) or from the System Interfaces volume of IEEE Std. 1003.1-200x environment inherited by the shell when it begins (see the *export* on page 111 special built-in)
- Shell functions; see Section 2.9.5 on page 79
- Options turned on at invocation or by *set*
- Process IDs of the last commands in asynchronous lists known to this shell environment; see Section 2.9.3.1 on page 74
- Shell aliases; see Section 2.3.1 on page 40

Utilities other than the special built-ins (see Section 2.14 on page 96) shall be invoked in a separate environment that consists of the following. The initial value of these objects shall be the same as that for the parent shell, except as noted below.

- Open files inherited on invocation of the shell, open files controlled by the *exec* special built-in plus any modifications, and additions specified by any redirections to the utility
- Current working directory
- File creation mask
- If the utility is a shell script, traps caught by the shell shall be set to the default values and traps ignored by the shell shall be set to be ignored by the utility; if the utility is not a shell script, the trap actions (default or ignore) shall be mapped into the appropriate signal handling actions for the utility
- Variables with the *export* attribute, along with those explicitly exported for the duration of the command, shall be passed to the utility as System Interfaces volume of IEEE Std. 1003.1-200x environment variables

The environment of the shell process shall not be changed by the utility unless explicitly specified by the utility description (for example, *cd* and *umask*).

A subshell environment shall be created as a duplicate of the shell environment, except that signal traps set by that shell environment shall be set to the default values. Changes made to the subshell environment shall not affect the shell environment. Command substitution, commands that are grouped with parentheses, and asynchronous lists shall be executed in a subshell environment. Additionally, each command of a multi-command pipeline is in a subshell environment; as an extension, however, any or all commands in a pipeline may be executed in the current environment. All other commands shall be executed in the current shell environment.

3378 **Rationale**

3379 Some systems have implemented the last stage of a pipeline in the current environment so that
3380 commands such as:

3381 `command | read foo`

3382 set variable **foo** in the current environment. This extension is allowed, but not required;
3383 therefore, a shell programmer should consider a pipeline to be in a subshell environment, but
3384 not depend on it.

3385 In early proposals, the description of execution environment failed to mention that each
3386 command in a multiple command pipeline could be in a subshell execution environment. For
3387 compatibility with some historical shells, the wording was phrased to allow an implementation
3388 to place any or all commands of a pipeline in the current environment. However, this means that
3389 a POSIX application must assume each command is in a subshell environment, but not depend
3390 on it.

3391 The wording about shell scripts is meant to convey the fact that describing “trap actions” can
3392 only be understood in the context of the shell command language. Outside of this context, such
3393 as in a C-language program, signals are the operative condition, not traps.

2.13 Pattern Matching Notation

The pattern matching notation described in this section is used to specify patterns for matching strings in the shell. Historically, pattern matching notation is related to, but slightly different from, the regular expression notation described in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 9, Regular Expressions. For this reason, the description of the rules for this pattern matching notation are based on the description of regular expression notation, modified to include backslash escape processing.

Rationale

Pattern matching is a simpler concept and has a simpler syntax than REs, as the former is generally used for the manipulation of file names, which are relatively simple collections of characters, while the latter is generally used to manipulate arbitrary text strings of potentially greater complexity. However, some of the basic concepts are the same, so this section points liberally to the detailed descriptions in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 9, Regular Expressions.

2.13.1 Patterns Matching a Single Character

The following *patterns matching a single character* match a single character: *ordinary characters*, *special pattern characters*, and *pattern bracket expressions*. The pattern bracket expression also shall match a single collating element. A backslash character shall escape the following character. The escaping backslash shall be discarded.

An ordinary character is a pattern that shall match itself. It can be any character in the supported character set except for NUL, those special shell characters in Section 2.2 on page 36 that require quoting, and the following three special pattern characters. Matching shall be based on the bit pattern used for encoding the character, not on the graphic representation of the character. If any character (ordinary, shell special, or pattern special) is quoted, that pattern shall match the character itself. The shell special characters always require quoting.

When unquoted and outside a bracket expression, the following three characters shall have special meaning in the specification of patterns:

- ? A question-mark is a pattern that shall match any character.
- * An asterisk is a pattern that shall match multiple characters, as described in Section 2.13.2 on page 93.
- [The open bracket shall introduce a pattern bracket expression.

The description of basic regular expression bracket expressions in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 9.3.5, RE Bracket Expression shall also apply to the pattern bracket expression, except that the exclamation-mark character (' ! ') shall replace the circumflex character (' ^ ') in its role in a *non-matching list* in the regular expression notation. A bracket expression starting with an unquoted circumflex character produces unspecified results.

When pattern matching is used where shell quote removal is not performed (such as in the argument to the *find name* primary when *find* is being called using one of the *exec* functions as defined in the System Interfaces volume of IEEE Std. 1003.1-200x, or in the *pattern* argument to the *fnmatch()* function), special characters can be escaped to remove their special meaning by preceding them with a backslash character. This escaping backslash is discarded. The sequence " \\ " represents one literal backslash. All of the requirements and effects of quoting on ordinary, shell special, and special pattern characters shall apply to escaping in this context.

Rationale

Both quoting and escaping are described here because pattern matching must work in three separate circumstances:

1. Calling directly upon the shell, such as in path name expansion or in a **case** statement. All of the following match the string or file **abc**:

```
abc "abc" a"b" c a\bc a[b]c a["b"]c a[\b]c a["\b"]c a?c a*c
```

The following do not:

```
"a?c" a*c a[b]c
```

2. Calling a utility or function without going through a shell, as described for *find* and the *fnmatch()* function defined in the System Interfaces volume of IEEE Std. 1003.1-200x.
3. Calling utilities such as *find*, *cpio*, *tar*, or *pax* through the shell command line. In this case, shell quote removal is performed before the utility sees the argument. For example, in:

```
find /bin -name "e\c[\h]o" -print
```

after quote removal, the backslashes are presented to *find* and it treats them as escape characters. Both precede ordinary characters, so the *c* and *h* represent themselves and *echo* would be found on many historical systems (that have it in */bin*). To find a file name that contained shell special characters or pattern characters, both quoting and escaping are required, such as:

```
pax -r ... "*a\(\?"
```

to extract a file name ending with "a(?".

Conforming applications are required to quote or escape the shell special characters (sometimes called metacharacters). If used without this protection, syntax errors can result or implementation extensions can be triggered. For example, the KornShell supports a series of extensions based on parentheses in patterns.

The restriction on a circumflex in a bracket expression is to allow implementations that support pattern matching using the circumflex as the negation character in addition to the exclamation-mark. A portable application must use something like "[^!]" to match either character.

2.13.2 Patterns Matching Multiple Characters

The following rules are used to construct *patterns matching multiple characters* from *patterns matching a single character*:

1. The asterisk ('*') is a pattern that shall match any string, including the null string.
2. The concatenation of *patterns matching a single character* is a valid pattern that shall match the concatenation of the single characters or collating elements matched by each of the concatenated patterns.
3. The concatenation of one or more *patterns matching a single character* with one or more asterisks is a valid pattern. In such patterns, each asterisk shall match a string of zero or more characters, matching the greatest possible number of characters that still allows the remainder of the pattern to match the string.

3476 **Rationale**

3477 Since each asterisk matches zero or more occurrences, the patterns "a*b" and "a**b" have
 3478 identical functionality.

3479 **Examples**

3480 a[bc] Matches the strings "ab" and "ac".

3481 a*d Matches the strings "ad", "abd", and "abcd", but not the string "abc".

3482 a*d* Matches the strings "ad", "abcd", "abcdef", "aaaad", and "adddd".

3483 *a*d Matches the strings "ad", "abcd", "efabcd", "aaaad", and "adddd".

3484 **2.13.3 Patterns Used for File Name Expansion**

3485 The rules described so far in Section 2.13.1 on page 92 and Section 2.13.2 on page 93 are qualified
 3486 by the following rules that apply when pattern matching notation is used for file name
 3487 expansion:

3488 1. The application shall ensure that the slash character in a path name is explicitly matched
 3489 by using one or more slashes in the pattern; it cannot be matched by the asterisk or
 3490 question-mark special characters or by a bracket expression. Slashes in the pattern are
 3491 identified before bracket expressions; thus, a slash cannot be included in a pattern bracket
 3492 MAN expression used for file name expansion. If a slash character is found following an
 3493 unescaped open square bracket character before a corresponding closing square bracket is
 3494 found, the open bracket is treated as an ordinary character. For example, the pattern
 3495 "a[b/c]d" does not match such path names as **abd** or **a/d**. It only matches a path name
 3496 of literally **a[b/c]d**.

3497 2. If a file name begins with a period ('.'), the application shall ensure that the period is
 3498 explicitly matched by using a period as the first character of the pattern or immediately
 3499 following a slash character. The leading period shall not be matched by:

- 3500 • The asterisk or question-mark special characters
- 3501 • A bracket expression containing a non-matching list, such as "[!a]", a range
- 3502 expression, such as "[%-0]", or a character class expression, such as "[[:punct:]]"

3503 It is unspecified whether an explicit period in a bracket expression matching list, such as
 3504 "[.abc]", can match a leading period in a file name.

3505 3. Specified patterns are matched against existing file names and path names, as appropriate.
 3506 Each component that contains a pattern character requires read permission in the directory
 3507 containing that component. Any component, except the last, that does not contain a
 3508 pattern character requires search permission. For example, given the pattern:

3509 /foo/bar/x*/bam

3510 search permission is needed for directories / and **foo**, search and read permissions are
 3511 needed for directory **bar**, and search permission is needed for each **x*** directory. If the
 3512 pattern matches any existing file names or path names, the pattern shall be replaced with
 3513 those file names and path names, sorted according to the collating sequence in effect in the
 3514 current locale. If the pattern contains an invalid bracket expression or does not match any
 3515 existing file names or path names, the pattern string shall be left unchanged.

Rationale

The caveat about a slash within a bracket expression is derived from historical practice. The pattern "a[b/c]d" does not match such path names as **abd** or **a/d**. On some systems (including those conforming to the Single UNIX Specification), it matched a path name of literally "a[b/c]d". On other systems, it produced an undefined condition (an unescaped '[' used outside a bracket expression). In this version, the XSI behavior is now required.

Filenames beginning with a period historically have been specially protected from view on UNIX systems. A proposal to allow an explicit period in a bracket expression to match a leading period was considered; it is allowed as an implementation extension, but a conforming application cannot make use of it. If this extension becomes popular in the future, it will be considered for a future version of this volume of IEEE Std. 1003.1-200x.

Historical systems have varied in their permissions requirements. To match **f*/bar** has required read permissions on the **f*** directories in the System V shell, but this volume of IEEE Std. 1003.1-200x, the C shell, and KornShell require only search permissions.

3530 2.14 Special Built-In Utilities

3531 The following *special built-in* utilities shall be supported in the shell command language. The
3532 output of each command, if any, shall be written to standard output, subject to the normal
3533 redirection and piping possible with all commands.

3534 The term *built-in* implies that the shell can execute the utility directly and does not need to
3535 search for it. An implementation can choose to make any utility a built-in; however, the special
3536 built-in utilities described here differ from regular built-in utilities in two respects:

- 3537 1. A syntax error in a special built-in utility may cause a shell executing that utility to abort,
3538 while a syntax error in a regular built-in utility shall not cause a shell executing that utility
3539 to abort. (See Section 2.8.1 on page 65 for the consequences of errors on interactive and
3540 non-interactive shells.) If a special built-in utility encountering a syntax error does not
3541 abort the shell, its exit value shall be non-zero.
- 3542 2. Variable assignments specified with special built-in utilities remain in effect after the
3543 built-in completes; this shall not be the case with a regular built-in or other utility.

3544 The special built-in utilities in this section need not be provided in a manner accessible via the
3545 *exec* family of functions defined in the System Interfaces volume of IEEE Std. 1003.1-200x.

3546 Some of the special built-ins are described as conforming to the System Interface Definitions
3547 volume of IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines. For those that are not,
3548 the requirement in Section 1.11 on page 25 that "—" be recognized as a first argument to be
3549 discarded does not apply and a portable application shall not use that argument.

3550 **NAME**

3551 break — exit from for, while, or until loop

3552 **SYNOPSIS**3553 break [*n*]3554 **DESCRIPTION**

3555 The *break* utility shall exit from the smallest enclosing **for**, **while**, or **until** loop, if any; or from the
3556 *n*th enclosing loop if *n* is specified. The value of *n* is an unsigned decimal integer greater than or
3557 equal to 1. The default shall be equivalent to *n*=1. If *n* is greater than the number of enclosing
3558 loops, the last enclosing loop shall be exited from. Execution shall continue with the command
3559 immediately following the loop.

3560 **OPTIONS**

3561 None.

3562 **OPERANDS**

3563 None.

3564 **STDIN**

3565 None.

3566 **INPUT FILES**

3567 None.

3568 **ENVIRONMENT VARIABLES**

3569 None.

3570 **ASYNCHRONOUS EVENTS**

3571 None.

3572 **STDOUT**

3573 None.

3574 **STDERR**

3575 None.

3576 **OUTPUT FILES**

3577 None.

3578 **EXTENDED DESCRIPTION**

3579 None.

3580 **EXIT STATUS**

3581 0 Successful completion.

3582 >0 The *n* value was not an unsigned decimal integer greater than or equal to 1.3583 **CONSEQUENCES OF ERRORS**

3584 None.

3585 **APPLICATION USAGE**

3586 None.

3587 **EXAMPLES**

```
3588       for i in * do
3589           if test -d "$i" then break fi done
```

3590 **RATIONALE**

3591 In early proposals, consideration was given to expanding the syntax of *break* and *continue* to refer
3592 to a label associated with the appropriate loop as a preferable alternative to the *n* method.
3593 However, this volume of IEEE Std. 1003.1-200x does reserve the namespace of command names
3594 ending with a colon. It is anticipated that a future implementation could take advantage of this
3595 and provide something like:

```
3596       outofloop: for i in a b c d e
3597           do
3598               for j in 0 1 2 3 4 5 6 7 8 9
3599               do
3600                   if test -r "${i}${j}"
3601                   then break outofloop
3602                   fi
3603               done
3604           done
```

3605 and that this might be standardized after implementation experience is achieved.

3606 **FUTURE DIRECTIONS**

3607 None.

3608 **SEE ALSO**

3609 Section 2.14 on page 96

3610 **CHANGE HISTORY**

3611 None.

3612 **NAME**

3613 colon — null utility

3614 **SYNOPSIS**3615 : [*argument* ...]3616 **DESCRIPTION**

3617 This utility shall only expand command *arguments*. It is used when a command is needed, as in
3618 the *then* condition of an *if* command, but nothing is to be done by the command.

3619 **OPTIONS**

3620 None.

3621 **OPERANDS**

3622 None.

3623 **STDIN**

3624 None.

3625 **INPUT FILES**

3626 None.

3627 **ENVIRONMENT VARIABLES**

3628 None.

3629 **ASYNCHRONOUS EVENTS**

3630 None.

3631 **STDOUT**

3632 None.

3633 **STDERR**

3634 None.

3635 **OUTPUT FILES**

3636 None.

3637 **EXTENDED DESCRIPTION**

3638 None.

3639 **EXIT STATUS**

3640 Zero.

3641 **CONSEQUENCES OF ERRORS**

3642 None.

3643 **APPLICATION USAGE**

3644 None.

3645 **EXAMPLES**

```
3646 : ${X=abc}  
3647 if     false  
3648 then  :  
3649 else  echo $X  
3650 fi  
3651 abc
```

3652 As with any of the special built-ins, the null utility can also have variable assignments and
3653 redirections associated with it, such as:

3654 `x=y : > z`

3655 which sets variable `x` to the value `y` (so that it persists after the null utility completes) and creates
3656 or truncates file `z`.

3657 **RATIONALE**

3658 None.

3659 **FUTURE DIRECTIONS**

3660 None.

3661 **SEE ALSO**

3662 Section 2.14 on page 96

3663 **CHANGE HISTORY**

3664 None.

3665 **NAME**

3666 continue — continue for, while, or until loop

3667 **SYNOPSIS**3668 continue [*n*]3669 **DESCRIPTION**

3670 The *continue* utility shall return to the top of the smallest enclosing **for**, **while**, or **until** loop, or to
3671 the top of the *n*th enclosing loop, if *n* is specified. This involves repeating the condition list of a
3672 **while** or **until** loop or performing the next assignment of a **for** loop, and reexecuting the loop if
3673 appropriate.

3674 The value of *n* is a decimal integer greater than or equal to 1. The default is equivalent to *n*=1. If
3675 *n* is greater than the number of enclosing loops, the last enclosing loop shall be used.

3676 **OPTIONS**

3677 None.

3678 **OPERANDS**

3679 None.

3680 **STDIN**

3681 None.

3682 **INPUT FILES**

3683 None.

3684 **ENVIRONMENT VARIABLES**

3685 None.

3686 **ASYNCHRONOUS EVENTS**

3687 None.

3688 **STDOUT**

3689 None.

3690 **STDERR**

3691 None.

3692 **OUTPUT FILES**

3693 None.

3694 **EXTENDED DESCRIPTION**

3695 None.

3696 **EXIT STATUS**

3697 0 Successful completion.

3698 >0 The *n* value was not an unsigned decimal integer greater than or equal to 1.3699 **CONSEQUENCES OF ERRORS**

3700 None.

3701 **APPLICATION USAGE**

3702 None.

3703 **EXAMPLES**

3704 for i in *

3705 do

3706 if test -d "\$i"

3707 then continue

3708 fi

3709 echo "\"\$i\" \" is not a directory.

3710 done

3711 **RATIONALE**

3712 None.

3713 **FUTURE DIRECTIONS**

3714 None.

3715 **SEE ALSO**

3716 Section 2.14 on page 96

3717 **CHANGE HISTORY**

3718 None.

3719 **NAME**
3720 dot — execute commands in current environment

3721 **SYNOPSIS**
3722 . *file*

3723 **DESCRIPTION**
3724 The shell shall execute commands from the *file* in the current environment.
3725 If *file* does not contain a slash, the shell shall use the search path specified by *PATH* to find the
3726 directory containing *file*. Unlike normal command search, however, the file searched for by the
3727 *dot* utility need not be executable. If no readable file is found, a non-interactive shell shall abort;
3728 an interactive shell shall write a diagnostic message to standard error, but this condition shall
3729 not be considered a syntax error.

3730 **OPTIONS**
3731 None.

3732 **OPERANDS**
3733 None.

3734 **STDIN**
3735 None.

3736 **INPUT FILES**
3737 None.

3738 **ENVIRONMENT VARIABLES**
3739 None.

3740 **ASYNCHRONOUS EVENTS**
3741 None.

3742 **STDOUT**
3743 None.

3744 **STDERR**
3745 None.

3746 **OUTPUT FILES**
3747 None.

3748 **EXTENDED DESCRIPTION**
3749 None.

3750 **EXIT STATUS**
3751 Returns the value of the last command executed, or a zero exit status if no command is executed.

3752 **CONSEQUENCES OF ERRORS**
3753 None.

3754 **APPLICATION USAGE**
3755 None.

3756 **EXAMPLES**
3757 cat foobar
3758 **foo=hello bar=world**
3759 . foobar
3760 echo \$foo \$bar
3761 **hello world**

3762 RATIONALE

3763 Some older implementations searched the current directory for the *file*, even if the value of *PATH*
3764 disallowed it. This behavior was omitted from this volume of IEEE Std. 1003.1-200x due to
3765 concerns about introducing the susceptibility to trojan horses that the user might be trying to
3766 avoid by leaving *dot* out of *PATH*.

3767 The KornShell version of *dot* takes optional arguments that are set to the positional parameters.
3768 This is a valid extension that allows a *dot* script to behave identically to a function.

3769 FUTURE DIRECTIONS

3770 None.

3771 SEE ALSO

3772 Section 2.14 on page 96

3773 CHANGE HISTORY

3774 None.

3775 **NAME**

3776 eval — construct command by concatenating arguments

3777 **SYNOPSIS**3778 eval [*argument* ...]3779 **DESCRIPTION**3780 The *eval* utility shall construct a command by concatenating *arguments* together, separating each
3781 with a <space> character. The constructed command shall be read and executed by the shell.3782 **OPTIONS**

3783 None.

3784 **OPERANDS**

3785 None.

3786 **STDIN**

3787 None.

3788 **INPUT FILES**

3789 None.

3790 **ENVIRONMENT VARIABLES**

3791 None.

3792 **ASYNCHRONOUS EVENTS**

3793 None.

3794 **STDOUT**

3795 None.

3796 **STDERR**

3797 None.

3798 **OUTPUT FILES**

3799 None.

3800 **EXTENDED DESCRIPTION**

3801 None.

3802 **EXIT STATUS**3803 If there are no *arguments*, or only null arguments, *eval* shall return a zero exit status; otherwise, it
3804 shall return the exit status of the command defined by the string of concatenated *arguments*
3805 separated by spaces.3806 **CONSEQUENCES OF ERRORS**

3807 None.

3808 **APPLICATION USAGE**

3809 None.

3810 **EXAMPLES**

3811 foo=10 x=foo

3812 y=' '\$x

3813 echo \$y

3814 \$foo

3815 eval y=' '\$x

3816 echo \$y

3817 10

3818 **RATIONALE**

3819 None.

3820 **FUTURE DIRECTIONS**

3821 None.

3822 **SEE ALSO**

3823 Section 2.14 on page 96

3824 **CHANGE HISTORY**

3825 None.

3826 **NAME**

3827 *exec* — execute commands and open, close, or copy file descriptors

3828 **SYNOPSIS**

3829 *exec* [*command* [*argument* ...]]

3830 **DESCRIPTION**

3831 The *exec* utility shall open, close, and/or copy file descriptors as specified by any redirections as
3832 part of the command.

3833 If *exec* is specified without *command* or *arguments*, and any file descriptors with numbers greater
3834 than 2 are opened with associated redirection statements, it is unspecified whether those file
3835 descriptors remain open when the shell invokes another utility. Scripts concerned that child
3836 shells could misuse open file descriptors can always close them explicitly, as shown in one of the
3837 following examples.

3838 If *exec* is specified with *command*, it shall replace the shell with *command* without creating a new
3839 process. If *arguments* are specified, they shall be arguments to *command*. Redirection affects the
3840 current shell execution environment.

3841 **OPTIONS**

3842 None.

3843 **OPERANDS**

3844 None.

3845 **STDIN**

3846 None.

3847 **INPUT FILES**

3848 None.

3849 **ENVIRONMENT VARIABLES**

3850 None.

3851 **ASYNCHRONOUS EVENTS**

3852 None.

3853 **STDOUT**

3854 None.

3855 **STDERR**

3856 None.

3857 **OUTPUT FILES**

3858 None.

3859 **EXTENDED DESCRIPTION**

3860 None.

3861 **EXIT STATUS**

3862 If *command* is specified, *exec* shall not return to the shell; rather, the exit status of the process shall
3863 be the exit status of the program implementing *command*, which overlaid the shell. If *command* is
3864 not found, the exit status shall be 127. If *command* is found, but it is not an executable utility, the
3865 exit status shall be 126. If a redirection error occurs (see Section 2.8.1 on page 65), the shell shall
3866 exit with a value in the range 1–125. Otherwise, *exec* shall return a zero exit status.

3867 CONSEQUENCES OF ERRORS

3868 None.

3869 APPLICATION USAGE

3870 None.

3871 EXAMPLES

3872 Open *readfile* as file descriptor 3 for reading:

3873 exec 3< readfile

3874 Open *writefile* as file descriptor 4 for writing:

3875 exec 4> writefile

3876 Make file descriptor 5 a copy of file descriptor 0:

3877 exec 5<&0

3878 Close file descriptor 3:

3879 exec 3<&-

3880 Cat the file **maggie** by replacing the current shell with the *cat* utility:

3881 exec cat maggie

3882 RATIONALE

3883 Most historical implementations were not conformant in that:

3884 foo=bar exec cmd

3885 did not pass **foo** to **cmd**.

3886 FUTURE DIRECTIONS

3887 None.

3888 SEE ALSO

3889 Section 2.14 on page 96

3890 CHANGE HISTORY

3891 None.

3892 **NAME**

3893 exit — cause the shell to exit

3894 **SYNOPSIS**3895 exit [*n*]3896 **DESCRIPTION**

3897 The *exit* utility shall cause the shell to exit with the exit status specified by the unsigned decimal
3898 integer *n*. If *n* is specified, but its value is not between 0 and 255 inclusively, the exit status is
3899 undefined.

3900 A *trap* on **EXIT** shall be executed before the shell terminates, except when the *exit* utility is
3901 invoked in that *trap* itself, in which case the shell shall exit immediately.

3902 **OPTIONS**

3903 None.

3904 **OPERANDS**

3905 None.

3906 **STDIN**

3907 None.

3908 **INPUT FILES**

3909 None.

3910 **ENVIRONMENT VARIABLES**

3911 None.

3912 **ASYNCHRONOUS EVENTS**

3913 None.

3914 **STDOUT**

3915 None.

3916 **STDERR**

3917 None.

3918 **OUTPUT FILES**

3919 None.

3920 **EXTENDED DESCRIPTION**

3921 None.

3922 **EXIT STATUS**

3923 The exit status shall be *n*, if specified. Otherwise, the value shall be the exit value of the last
3924 command executed, or zero if no command was executed. When *exit* is executed in a *trap* action,
3925 the last command is considered to be the command that executed immediately preceding the
3926 *trap* action.

3927 **CONSEQUENCES OF ERRORS**

3928 None.

3929 **APPLICATION USAGE**

3930 None.

3931 **EXAMPLES**3932 Exit with a *true* value:

3933 exit 0

3934 Exit with a *false* value:

3935 exit 1

3936 **RATIONALE**

3937 As explained in other sections, certain exit status values have been reserved for special uses and
3938 should be used by applications only for those purposes:

3939 126 A file to be executed was found, but it was not an executable utility.

3940 127 A utility to be executed was not found.

3941 >128 A command was interrupted by a signal.

3942 **FUTURE DIRECTIONS**

3943 None.

3944 **SEE ALSO**

3945 Section 2.14 on page 96

3946 **CHANGE HISTORY**

3947 None.

3948 **NAME**

3949 export — set export attribute for variables

3950 **SYNOPSIS**

3951 export name[=word]...

3952 export -p

3953 **DESCRIPTION**3954 The shell shall give the export attribute to the variables corresponding to the specified *names*,
3955 which shall cause them to be in the environment of subsequently executed commands.3956 The *export* special built-in shall support the System Interface Definitions volume of
3957 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.3958 When **-p** is specified, *export* shall write to the standard output the names and values of all
3959 exported variables, in the following format:

3960 "export %s=%s\n", <name>, <value>

3961 The shell shall format the output, including the proper use of quoting, so that it is suitable for
3962 reinput to the shell as commands that achieve the same exporting results.

3963 When no arguments are given, the results are unspecified.

3964 **OPTIONS**

3965 None.

3966 **OPERANDS**

3967 None.

3968 **STDIN**

3969 None.

3970 **INPUT FILES**

3971 None.

3972 **ENVIRONMENT VARIABLES**

3973 None.

3974 **ASYNCHRONOUS EVENTS**

3975 None.

3976 **STDOUT**

3977 None.

3978 **STDERR**

3979 None.

3980 **OUTPUT FILES**

3981 None.

3982 **EXTENDED DESCRIPTION**

3983 None.

3984 **EXIT STATUS**

3985 Zero.

3986 **CONSEQUENCES OF ERRORS**

3987 None.

3988 **APPLICATION USAGE**

3989 None.

3990 **EXAMPLES**3991 Export *PWD* and *HOME* variables:3992 `export PWD HOME`3993 Set and export the *PATH* variable:3994 `export PATH=/local/bin:$PATH`

3995 Save and restore all exported variables:

3996 `export -p > temp-file`3997 `unset a lot of variables`3998 `... processing`3999 `. temp-file`4000 **RATIONALE**

4001 Some historical shells use the no-argument case as the functional equivalent of what is required
4002 here with **-p**. This feature was left unspecified because it is not historical practice in all shells,
4003 and some scripts may rely on the now-unspecified results on their implementations. Attempts to
4004 specify the **-p** output as the default case were unsuccessful in achieving consensus. The **-p**
4005 option was added to allow portable access to the values that can be saved and then later restored
4006 using; for example, a *dot* script.

4007 **FUTURE DIRECTIONS**

4008 None.

4009 **SEE ALSO**

4010 Section 2.14 on page 96

4011 **CHANGE HISTORY**

4012 None.

4013 **NAME**

4014 readonly — set read-only attribute for variables

4015 **SYNOPSIS**

4016 readonly name[=word]...

4017 readonly -p

4018 **DESCRIPTION**

4019 The variables whose *names* are specified shall be given the *readonly* attribute. The values of
 4020 variables with the *readonly* attribute cannot be changed by subsequent assignment, nor can those
 4021 variables be unset by the *unset* utility.

4022 The *readonly* special built-in shall support the System Interface Definitions volume of
 4023 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

4024 When **-p** is specified, *readonly* writes to the standard output the names and values of all read-
 4025 only variables, in the following format:

4026 "readonly %s=%s\n", <name>, <value>

4027 The shell shall format the output, including the proper use of quoting, so that it is suitable for
 4028 reinput to the shell as commands that achieve the same attribute-setting results.

4029 When no arguments are given, the results are unspecified.

4030 **OPTIONS**

4031 None.

4032 **OPERANDS**

4033 None.

4034 **STDIN**

4035 None.

4036 **INPUT FILES**

4037 None.

4038 **ENVIRONMENT VARIABLES**

4039 None.

4040 **ASYNCHRONOUS EVENTS**

4041 None.

4042 **STDOUT**

4043 None.

4044 **STDERR**

4045 None.

4046 **OUTPUT FILES**

4047 None.

4048 **EXTENDED DESCRIPTION**

4049 None.

4050 **EXIT STATUS**

4051 Zero.

4052 **CONSEQUENCES OF ERRORS**

4053 None.

4054 **APPLICATION USAGE**

4055 None.

4056 **EXAMPLES**4057 `readonly HOME PWD`4058 **RATIONALE**

4059 Some historical shells preserve the read-only attribute across separate invocations. This volume
4060 of IEEE Std. 1003.1-200x allows this behavior, but does not require it.

4061 The `-p` option allows portable access to the values that can be saved and then later restored
4062 using; for example, a *dot* script. Also see the RATIONALE for *export* on page 111 for a
4063 description of the no-argument and `-p` output cases and a related example.

4064 Read-only functions were considered, but they were omitted as not being historical practice or
4065 particularly useful. Furthermore, functions must not be *readonly* across invocations to preclude
4066 *spoofing* (spoofing is the term for the practice of creating a program that acts like a well-known
4067 utility with the intent of subverting the real intent of the user) of administrative or security-
4068 relevant (or security-conscious) shell scripts.

4069 **FUTURE DIRECTIONS**

4070 None.

4071 **SEE ALSO**

4072 Section 2.14 on page 96

4073 **CHANGE HISTORY**

4074 None.

4075 **NAME**

4076 return — return from a function

4077 **SYNOPSIS**4078 return [*n*]4079 **DESCRIPTION**

4080 The *return* utility shall cause the shell to stop executing the current function or *dot* script. If the
4081 shell is not currently executing a function or *dot* script, the results are unspecified.

4082 **OPTIONS**

4083 None.

4084 **OPERANDS**

4085 None.

4086 **STDIN**

4087 None.

4088 **INPUT FILES**

4089 None.

4090 **ENVIRONMENT VARIABLES**

4091 None.

4092 **ASYNCHRONOUS EVENTS**

4093 None.

4094 **STDOUT**

4095 None.

4096 **STDERR**

4097 None.

4098 **OUTPUT FILES**

4099 None.

4100 **EXTENDED DESCRIPTION**

4101 None.

4102 **EXIT STATUS**

4103 The value of the special parameter '*?*' shall be set to *n*, an unsigned decimal integer, or to the
4104 exit status of the last command executed if *n* is not specified. If the value of *n* is greater than 255,
4105 the results are undefined. When *return* is executed in a *trap* action, the last command is
4106 considered to be the command that executed immediately preceding the *trap* action.

4107 **CONSEQUENCES OF ERRORS**

4108 None.

4109 **APPLICATION USAGE**

4110 None.

4111 **EXAMPLES**

4112 None.

4113 **RATIONALE**

4114 The behavior of *return* when not in a function or *dot* script differs between the System V shell
4115 and the KornShell. In the System V shell this is an error, whereas in the KornShell, the effect is
4116 the same as *exit*.

4117 The results of returning a number greater than 255 are undefined because of differing practices
4118 in the various historical implementations. Some shells AND out all but the low-order 8 bits;
4119 others allow larger values, but not of unlimited size.

4120 See the discussion of appropriate exit status values under *exit* on page 109.

4121 **FUTURE DIRECTIONS**

4122 None.

4123 **SEE ALSO**

4124 Section 2.14 on page 96

4125 **CHANGE HISTORY**

4126 None.

4127 **NAME**

4128 set — set or unset options and positional parameters

4129 **SYNOPSIS**4130 xSI set [-abCefmnuvx] [-h] [-o *option*] [*argument*...]4131 xSI set [+abCefmnuvx] [+h] [+o *option*] [*argument*...]4132 set — [*argument*...]

4133 set -o

4134 set +o

4135 **DESCRIPTION**

4136 If no options or *arguments* are specified, *set* shall write the names and values of all shell variables in the collation sequence of the current locale. Each *name* shall start on a separate line, using the format:

4139 "%s=%s\n", <name>, <value>

4140 The *value* string shall be written with appropriate quoting so that it is suitable for reinput to the shell, setting or resetting, as far as possible, the variables that are currently set. Read-only variables cannot be reset; see the description of shell quoting in Section 2.2 on page 36.

4143 When options are specified, they shall set or unset attributes of the shell, as described below. When *arguments* are specified, they cause positional parameters to be set or unset, as described below. Setting or unsetting attributes and positional parameters are not necessarily related actions, but they can be combined in a single invocation of *set*.

4147 The *set* special built-in shall support the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines except that options can be specified with either a leading hyphen (meaning enable the option) or plus sign (meaning disable it).

4150 Implementations shall support the options in the following list in both their hyphen and plus-sign forms. These options can also be specified as options to *sh*.

4152 **-a** When this option is on, the export attribute shall be set for each variable to which an assignment is performed; see the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 3.426, Variable Assignment. If the assignment precedes a utility name in a command, the export attribute shall not persist in the current execution environment after the utility completes, with the exception that preceding one of the special built-in utilities causes the export attribute to persist after the built-in has completed. If the assignment does not precede a utility name in the command, or if the assignment is a result of the operation of the *getopts* or *read* utilities, the export attribute shall persist until the variable is unset.

4161 **-b** This option is supported if the system supports the User Portability Utilities option. It shall cause the shell to notify the user asynchronously of background job completions. The following message is written to standard error:

4164 "%d] %c %s %s\n", <job-number>, <current>, <status>, <job-name>

4165 where the fields shall be as follows:

4166 <current> The character '+' identifies the job that would be used as a default for the *fg* or *bg* utilities; this job can also be specified using the *job_id* "%+" or "%%". The character '-' identifies the job that would become the default if the current default job were to exit; this job can also be specified using the *job_id* "%-". For other jobs, this field is a <space> character. At most

one job can be identified with '+' and at most one job can be identified with '-'. If there is any suspended job, then the current job shall be a suspended job. If there are at least two suspended jobs, then the previous job also shall be a suspended job.

<*job-number*> A number that can be used to identify the process group to the *wait*, *fg*, *bg*, and *kill* utilities. Using these utilities, the job can be identified by prefixing the job number with '% '.

<*status*> Unspecified.

<*job-name*> Unspecified.

When the shell notifies the user a job has been completed, it may remove the job's process ID from the list of those known in the current shell execution environment; see Section 2.9.3.1 on page 74. Asynchronous notification shall not be enabled by default.

-C (Uppercase C.) Prevent existing files from being overwritten by the shell's '>' redirection operator (see Section 2.7.2 on page 61); the ">|" redirection operator shall override this *noclobber* option for an individual file.

-e When this option is on, if a simple command fails for any of the reasons listed in Section 2.8.1 on page 65 or returns an exit status value >0, and is not part of the compound list following a **while**, **until**, or **if** keyword, and is not a part of an AND or OR list, and is not a pipeline preceded by the **!** reserved word, then the shell shall immediately exit.

-f The shell shall disable path name expansion.

XSI -h Locate and remember utilities invoked by functions as those functions are defined (the utilities are normally located when the function is executed).

-m This option is supported if the system supports the User Portability Utilities option. All jobs shall be run in their own process groups. Immediately before the shell issues a prompt after completion of the background job, a message reporting the exit status of the background job shall be written to standard error. If a foreground job stops, the shell shall write a message to standard error to that effect, formatted as described by the *jobs* utility. In addition, if a job changes status other than exiting (for example, if it stops for input or output or is stopped by a SIGSTOP signal), the shell shall write a similar message immediately prior to writing the next prompt. This option is enabled by default for interactive shells.

-n The shell shall read commands but does not execute them; this can be used to check for shell script syntax errors. An interactive shell may ignore this option.

-o Write the current settings of the options to standard output in an unspecified format.

+o Write the current option settings to standard output in a format that is suitable for reinput to the shell as commands that achieve the same options settings.

-o *option*
This option is supported if the system supports the User Portability Utilities option. It shall set various options, many of which shall be equivalent to the single option letters. The following values of *option* shall be supported:

allexport Equivalent to -a.

errexit Equivalent to -e.

ignoreeof Prevent an interactive shell from exiting on end-of-file. This setting prevents accidental logouts when control-D is entered. A user shall explicitly *exit* to leave the interactive shell.

4215	<i>monitor</i>	Equivalent to -m . This option is supported if the system supports the User Portability Utilities option.
4216		
4217	<i>noclobber</i>	Equivalent to -C (uppercase C).
4218	<i>noglob</i>	Equivalent to -f .
4219	<i>noexec</i>	Equivalent to -n .
4220	MAN <i>nolog</i>	Prevent the entry of function definitions into the command history; see Command History List on page 892.
4221		
4222	<i>notify</i>	Equivalent to -b .
4223	<i>nounset</i>	Equivalent to -u .
4224	<i>verbose</i>	Equivalent to -v .
4225	<i>vi</i>	Allow shell command line editing using the built-in <i>vi</i> editor. Enabling <i>vi</i> mode shall disable any other command line editing mode provided as an implementation extension.
4226		
4227		
4228		It need not be possible to set <i>vi</i> mode on for certain block-mode terminals.
4229	<i>xtrace</i>	Equivalent to -x .
4230	-u	The shell writes a message to standard error when it tries to expand a variable that is not set and immediately exit. An interactive shell shall not exit.
4231		
4232	-v	The shell writes its input to standard error as it is read.
4233	-x	The shell writes to standard error a trace for each command after it expands the command and before it executes it. It is unspecified whether the command that turns tracing off is traced.
4234		
4235		
4236		The default for all these options is off (unset) unless the shell was invoked with them on; see <i>sh</i> .
4237		The remaining arguments shall be assigned in order to the positional parameters. The special parameter ' # ' shall be set to reflect the number of positional parameters. All positional parameters shall be unset before any new values are assigned.
4238		
4239		
4240		The special argument " — " immediately following the <i>set</i> command name can be used to delimit the arguments if the first argument begins with ' + ' or ' - ', or to prevent inadvertent listing of all shell variables when there are no arguments. The command <i>set--</i> without <i>argument</i> shall unset all positional parameters and set the special parameter ' # ' to zero.
4241		
4242		
4243		
4244	OPTIONS	
4245	None.	
4246	OPERANDS	
4247	None.	
4248	STDIN	
4249	None.	
4250	INPUT FILES	
4251	None.	
4252	ENVIRONMENT VARIABLES	
4253	None.	

4254 **ASYNCHRONOUS EVENTS**

4255 None.

4256 **STDOUT**

4257 None.

4258 **STDERR**

4259 None.

4260 **OUTPUT FILES**

4261 None.

4262 **EXTENDED DESCRIPTION**

4263 None.

4264 **EXIT STATUS**

4265 Zero.

4266 **CONSEQUENCES OF ERRORS**

4267 None.

4268 **APPLICATION USAGE**

4269 None.

4270 **EXAMPLES**

4271 Write out all variables and their values:

4272 set

4273 Set \$1, \$2, and \$3 and set "\$#" to 3:

4274 set c a b

4275 Turn on the **-x** and **-v** options:

4276 set -xv

4277 Unset all positional parameters:

4278 set --

4279 Set \$1 to the value of **-x**, even if **x** begins with **'-'** or **'+'**:

4280 set -- "\$x"

4281 Set the positional parameters to the expansion of **x**, even if **x** expands with a leading **'-'** or **'+'**:

4282 set -- \$x

4283 **RATIONALE**

4284 The **set --** form is listed specifically in the SYNOPSIS even though this usage is implied by the
4285 Utility Syntax Guidelines. The explanation of this feature removes any ambiguity about whether
4286 the **set --** form might be misinterpreted as being equivalent to **set** without any options or
4287 arguments. The functionality of this form has been adopted from the KornShell. In System V, **set**
4288 **--** only unsets parameters if there is at least one argument; the only way to unset all parameters
4289 is to use *shift*. Using the KornShell version should not affect System V scripts because there
4290 should be no reason to issue it without arguments deliberately; if it were issued as, for example:

4291 set -- "\$@"

4292 and there were in fact no arguments resulting from "\$@", unsetting the parameters would have
4293 no result.

The *set* + form in early proposals was omitted as being an unnecessary duplication of *set* alone and not widespread historical practice.

The *noclobber* option was changed to allow *set -C* as well as the *set -o noclobber* option. The single-letter version was added so that the historical "\$-" paradigm would not be broken; see Section 2.5.2 on page 43.

The *-h* flag is related to command name hashing and is only required on XSI-conformant systems.

The following *set* flags were omitted intentionally with the following rationale:

-k The *-k* flag was originally added by the author of the Bourne shell to make it easier for users of pre-release versions of the shell. In early versions of the Bourne shell the construct *set name=value*, had to be used to assign values to shell variables. The problem with *-k* is that the behavior affects parsing, virtually precluding writing any compilers. To explain the behavior of *-k*, it is necessary to describe the parsing algorithm, which is implementation-dependent. For example:

```
set -k; echo name=value
```

and:

```
set x--k
echo name=value
```

behave differently. The interaction with functions is even more complex. What is more, the *-k* flag is never needed, since the command line could have been reordered.

-t The *-t* flag is hard to specify and almost never used. The only known use could be done with here-documents. Moreover, the behavior with *ksh* and *sh* differs. The reference page says that it exits after reading and executing one command. What is one command? If the input is *date;date*, *sh* executes both *date* commands while *ksh* does only the first.

Consideration was given to rewriting *set* to simplify its confusing syntax. A specific suggestion was that the *unset* utility should be used to unset options instead of using the non-*getopt()*-able *+option* syntax. However, the conclusion was reached that the historical practice of using *+option* was satisfactory and that there was no compelling reason to modify such widespread historical practice.

The *-o* option was adopted from the KornShell to address user needs. In addition to its generally friendly interface, *-o* is needed to provide the *vi* command line editing mode, for which historical practice yields no single-letter option name. (Although it might have been possible to invent such a letter, it was recognized that other editing modes would be developed and *-o* provides ample name space for describing such extensions.)

Historical implementations are inconsistent in the format used for *-o* option status reporting. The *+o* format without an option-argument was added to allow portable access to the options that can be saved and then later restored using, for instance, a dot script.

Historically, *sh* did trace the command *set +x*, but *ksh* did not.

It is not possible to use the "\$-" special parameter to determine the current setting of either of these two modes. In the KornShell, the *option* option-argument can be omitted, producing a report of current option settings. Since it breaks the Utility Syntax Guidelines, and since the output format was unspecified (it changed between KornShell versions), this usage was omitted.

The *ignoreeof* setting prevents accidental logouts when the end-of-file character (typically control-D) is entered. A user shall explicitly *exit* to leave the interactive shell.

4338 The *set -m* option was added to apply only to the UPE because it applies primarily to interactive
4339 use, not shell script applications.

4340 The ability to do asynchronous notification became available in the 1988 version of the
4341 KornShell. To have it occur, the user had to issue the command:

```
4342 trap "jobs -n" CLD
```

4343 The C shell provides two different levels of an asynchronous notification capability. The
4344 environment variable *notify* is analogous to what is done in *set -b* or *set -o notify*. When set, it
4345 notifies the user immediately of background job completions. When unset, this capability is
4346 turned off.

4347 The other notification ability comes through the built-in utility *notify*. The syntax is:

```
4348 notify [%job ... ]
```

4349 By issuing *notify* with no operands, it causes the C shell to notify the user asynchronously when
4350 the state of the current job changes. If given operands, *notify* asynchronously informs the user of
4351 changes in the states of the specified jobs.

4352 To add asynchronous notification to the POSIX shell, neither the KornShell extensions to *trap*,
4353 nor the C shell *notify* environment variable seemed appropriate (*notify* is not a proper POSIX
4354 environment variable name).

4355 The *set -b* option was selected as a compromise.

4356 The *notify* built-in was considered to have more functionality than was required for simple
4357 asynchronous notification.

4358 FUTURE DIRECTIONS

4359 None.

4360 SEE ALSO

4361 Section 2.14 on page 96

4362 CHANGE HISTORY

4363 Issue 6

4364 The obsolescent *set* command name followed by *'-'* has been removed.

4365 The following new requirements on POSIX implementations derive from alignment with the
4366 Single UNIX Specification:

- 4367 • The *nolog* option is added to *set -o*.

4368 **NAME**

4369 shift — shift positional parameters

4370 **SYNOPSIS**4371 shift [*n*]4372 **DESCRIPTION**

4373 The positional parameters shall be shifted. Positional parameter 1 shall be assigned the value of
4374 parameter (1+*n*), parameter 2 shall be assigned the value of parameter (2+*n*), and so on. The
4375 parameters represented by the numbers "\$#" down to "\$#-*n*+1" shall be unset, and the
4376 parameter '#' is updated to reflect the new number of positional parameters.

4377 The value *n* shall be an unsigned decimal integer less than or equal to the value of the special
4378 parameter '#'. If *n* is not given, it shall be assumed to be 1. If *n* is 0, the positional and special
4379 parameters are not changed.

4380 **OPTIONS**

4381 None.

4382 **OPERANDS**

4383 None.

4384 **STDIN**

4385 None.

4386 **INPUT FILES**

4387 None.

4388 **ENVIRONMENT VARIABLES**

4389 None.

4390 **ASYNCHRONOUS EVENTS**

4391 None.

4392 **STDOUT**

4393 None.

4394 **STDERR**

4395 None.

4396 **OUTPUT FILES**

4397 None.

4398 **EXTENDED DESCRIPTION**

4399 None.

4400 **EXIT STATUS**4401 The exit status is >0 if *n*>\$#; otherwise, it is zero.4402 **CONSEQUENCES OF ERRORS**

4403 None.

4404 **APPLICATION USAGE**

4405 None.

4406 **EXAMPLES**

4407 \$ set a b c d e

4408 \$ shift 2

4409 \$ echo \$*

4410 c d e

4411 **RATIONALE**

4412 None.

4413 **FUTURE DIRECTIONS**

4414 None.

4415 **SEE ALSO**

4416 Section 2.14 on page 96

4417 **CHANGE HISTORY**

4418 None.

4419 **NAME**

4420 times — write process times

4421 **SYNOPSIS**

4422 MAN times

4423

4424 **DESCRIPTION**

4425 Write the accumulated user and system times for the shell and for all of its child processes, in the
 4426 following POSIX locale format:

```
4427      "%dm%fs %dm%fs\n%dm%fs %dm%fs\n", <shell user minutes>,
4428      <shell user seconds>, <shell system minutes>,
4429      <shell system seconds>, <children user minutes>,
4430      <children user seconds>, <children system minutes>,
4431      <children system seconds>
```

4432 The four pairs of times correspond to the members of the **<sys/times.h> tms** structure (defined
 4433 in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 13, Headers) as
 4434 returned by *times()*: *tms_utime*, *tms_stime*, *tms_cutime*, and *tms_cstime*, respectively.

4435 **OPTIONS**

4436 None.

4437 **OPERANDS**

4438 None.

4439 **STDIN**

4440 None.

4441 **INPUT FILES**

4442 None.

4443 **ENVIRONMENT VARIABLES**

4444 None.

4445 **ASYNCHRONOUS EVENTS**

4446 None.

4447 **STDOUT**

4448 None.

4449 **STDERR**

4450 None.

4451 **OUTPUT FILES**

4452 None.

4453 **EXTENDED DESCRIPTION**

4454 None.

4455 **EXIT STATUS**

4456 Zero.

4457 **CONSEQUENCES OF ERRORS**

4458 None.

4459 APPLICATION USAGE

4460 None.

4461 EXAMPLES

4462 \$ times

4463 0m0.43s 0m1.11s

4464 8m44.18s 1m43.23s

4465 RATIONALE

4466 The *times* special built-in from the Single UNIX Specification is now required for all conforming
4467 shells.

4468 FUTURE DIRECTIONS

4469 None.

4470 SEE ALSO

4471 Section 2.14 on page 96

4472 CHANGE HISTORY

4473 None.

4474 **NAME**

4475 trap — trap signals

4476 **SYNOPSIS**4477 trap [*action condition ...*]4478 **DESCRIPTION**

4479 If *action* is '-', the shell shall reset each *condition* to the default value. If *action* is null (" "), the
 4480 shell shall ignore each specified *condition* if it arises. Otherwise, the argument *action* shall be read
 4481 and executed by the shell when one of the corresponding conditions arises. The action of *trap*
 4482 shall override a previous action (either default action or one explicitly set). The value of "\$?"
 4483 after the *trap* action completes shall be the value it had before *trap* was invoked.

4484 The condition can be EXIT, 0 (equivalent to EXIT), or a signal specified using a symbolic name,
 4485 without the SIG prefix, as listed in the tables of signal names in the <signal.h> header defined in
 4486 the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 13, Headers; for
 4487 example, HUP, INT, QUIT, TERM. Implementations may permit lowercase signal names or
 4488 names with the SIG prefix as an extension. Setting a trap for SIGKILL or SIGSTOP produces
 4489 undefined results.

4490 The environment in which the shell executes a *trap* on EXIT shall be identical to the environment
 4491 immediately after the last command executed before the *trap* on EXIT was taken.

4492 Each time *trap* is invoked, the *action* argument shall be processed in a manner equivalent to:

4493 eval "\$action"

4494 Signals that were ignored on entry to a non-interactive shell cannot be trapped or reset, although
 4495 no error need be reported when attempting to do so. An interactive shell may reset or catch
 4496 signals ignored on entry. Traps shall remain in place for a given shell until explicitly changed
 4497 with another *trap* command.

4498 When a subshell is entered, traps that are not being ignored are set to the default actions. This
 4499 does not imply that the *trap* command cannot be used within the subshell to set new traps.

4500 The *trap* command with no arguments shall write to standard output a list of commands
 4501 associated with each condition. The format shall be:

4502 "trap — %s %s ...\\n", <action>, <condition> ...

4503 The shell shall format the output, including the proper use of quoting, so that it is suitable for
 4504 reinput to the shell as commands that achieve the same trapping results. For example:

4505 save_traps=\$(trap)

4506 ...

4507 eval "\$save_traps"

4508 XSI XSI-conformant systems also allow numeric signal numbers for the conditions corresponding to
 4509 the following signal names:

4510
4511
4512 XSI
4513
4514
4515
4516
4517
4518

Signal Number	Signal Name
1	SIGHUP
2	SIGINT
3	SIGQUIT
6	SIGABRT
9	SIGKILL
14	SIGALRM
15	SIGTERM

4519 The *trap* special built-in shall conform to the System Interface Definitions volume of
4520 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

4521 OPTIONS

4522 None.

4523 OPERANDS

4524 None.

4525 STDIN

4526 None.

4527 INPUT FILES

4528 None.

4529 ENVIRONMENT VARIABLES

4530 None.

4531 ASYNCHRONOUS EVENTS

4532 None.

4533 STDOUT

4534 None.

4535 STDERR

4536 None.

4537 OUTPUT FILES

4538 None.

4539 EXTENDED DESCRIPTION

4540 None.

4541 EXIT STATUS

4542 XSI If the trap name or number is invalid, a non-zero exit status shall be returned; otherwise, zero
4543 XSI shall be returned. For both interactive and non-interactive shells, invalid signal names or
4544 numbers shall not be considered a syntax error and do not cause the shell to abort.

4545 CONSEQUENCES OF ERRORS

4546 None.

4547 APPLICATION USAGE

4548 None.

4549 EXAMPLES

4550 Write out a list of all traps and actions:

4551 trap

4552 Set a trap so the *logout* utility in the directory referred to by the *HOME* environment variable
4553 executes when the shell terminates:

4554 trap '\$HOME/logout' EXIT

4555 or:

4556 trap '\$HOME/logout' 0

4557 Unset traps on INT, QUIT, TERM, and EXIT:

4558 trap - INT QUIT TERM EXIT

4559 **RATIONALE**

4560 Implementations may permit lowercase signal names as an extension. Implementations may
4561 also accept the names with the SIG prefix; no known historical shell does so. The *trap* and *kill*
4562 utilities in this volume of IEEE Std. 1003.1-200x are now consistent in their omission of the SIG
4563 prefix for signal names. Some *kill* implementations do not allow the prefix, and *kill -l* lists the
4564 signals without prefixes.

4565 Trapping SIGKILL or SIGSTOP is syntactically accepted by some historical implementations, but
4566 it has no effect. Portable POSIX applications cannot attempt to trap these signals.

4567 The output format is not historical practice. Since the output of historical *trap* commands is not
4568 portable (because numeric signal values are not portable) and had to change to become so, an
4569 opportunity was taken to format the output in a way that a shell script could use to save and
4570 then later reuse a trap if it wanted.

4571 The KornShell uses an **ERR** trap that is triggered whenever *set -e* would cause an exit. This is
4572 allowable as an extension, but was not mandated, as other shells have not used it.

4573 The text about the environment for the EXIT trap invalidates the behavior of some historical
4574 versions of interactive shells which, for example, close the standard input before executing a
4575 trap on 0. For example, in some historical interactive shell sessions the following trap on 0 would
4576 always print "—":

4577 trap 'read foo; echo "\$foo—" ' 0

4578 **FUTURE DIRECTIONS**

4579 None.

4580 **SEE ALSO**

4581 Section 2.14 on page 96

4582 **CHANGE HISTORY**

4583 **Issue 6**

4584 XSI-conforming implementations provide the mapping of signal names to numbers given above
4585 (previously this had been marked obsolescent). Other implementations need not provide this
4586 optional mapping.

4587 **NAME**

4588 unset — unset values and attributes of variables and functions

4589 **SYNOPSIS**4590 unset [-fv] *name* ...4591 **DESCRIPTION**4592 Each variable or function specified by *name* shall be unset.4593 If **-v** is specified, *name* refers to a variable name and the shell shall unset it and remove it from
4594 the environment. Read-only variables cannot be unset.4595 If **-f** is specified, *name* refers to a function and the shell shall unset the function definition.4596 If neither **-f** nor **-v** is specified, *name* refers to a variable; if a variable by that name does not
4597 exist, it is unspecified whether a function by that name, if any, shall be unset.4598 Unsetting a variable or function that was not previously set shall not be considered an error and
4599 does not cause the shell to abort.4600 The *unset* special built-in shall support the System Interface Definitions volume of
4601 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

4602 Note that:

4603 VARIABLE=

4604 is not equivalent to an *unset* of **VARIABLE**; in the example, **VARIABLE** is set to " ". Also, the
4605 variables that can be *unset* should not be misinterpreted to include the special parameters (see
4606 Section 2.5.2 on page 43).4607 **OPTIONS**

4608 None.

4609 **OPERANDS**

4610 None.

4611 **STDIN**

4612 None.

4613 **INPUT FILES**

4614 None.

4615 **ENVIRONMENT VARIABLES**

4616 None.

4617 **ASYNCHRONOUS EVENTS**

4618 None.

4619 **STDOUT**

4620 None.

4621 **STDERR**

4622 None.

4623 **OUTPUT FILES**

4624 None.

4625 **EXTENDED DESCRIPTION**

4626 None.

4627 **EXIT STATUS**4628 0 All *name* operands were successfully unset.4629 >0 At least one *name* could not be unset.4630 **CONSEQUENCES OF ERRORS**

4631 None.

4632 **APPLICATION USAGE**

4633 None.

4634 **EXAMPLES**4635 Unset *VISUAL* variable:

4636 unset -v VISUAL

4637 Unset the functions **foo** and **bar**:

4638 unset -f foo bar

4639 **RATIONALE**4640 Consideration was given to omitting the **-f** option in favor of an *unfunction* utility, but the
4641 standard developers decided to retain historical practice.4642 The **-v** option was introduced because System V historically used one name space for both
4643 variables and functions. When *unset* is used without options, System V historically unset either a
4644 function or a variable, and there was no confusion about which one was intended. A portable
4645 POSIX application can use *unset* without an option to unset a variable, but not a function; the **-f**
4646 option must be used.4647 **FUTURE DIRECTIONS**

4648 None.

4649 **SEE ALSO**

4650 Section 2.14 on page 96

4651 **CHANGE HISTORY**

4652 None.

Batch Environment Services and Utilities

4654

4655 BE This chapter describes the services and utilities that shall be implemented on all systems that
 4656 claim conformance to the Batch Environment option.

4657 Notes to Reviewers

4658 *This section with side shading will not appear in the final copy. - Ed.*

4659 This text and the associated reference pages are imported from IEEE Std. 1003.2d-1994. A
 4660 suggestion received has been to change the name to Batch Services and the abbreviated margin
 4661 marker to BS.

4662 3.1 General Concepts

4663 3.1.1 Batch Client-Server Interaction

4664 Batch jobs are created and managed by batch servers. A batch client interacts with a batch server
 4665 to access batch services on behalf of the user. In order to use batch services, a user must have
 4666 access to a batch client.

4667 A batch server is a computational entity, such as a daemon process, that provides batch services.
 4668 Batch servers route, queue, modify, and execute batch jobs on behalf of batch clients.

4669 The batch utilities described in this volume of IEEE Std. 1003.1-200x (and listed in Table 3-1 on
 4670 page 134) are clients of batch services; they allow users to perform actions on the job such as
 4671 creating, modifying, and deleting batch jobs from a shell command line. Although these batch
 4672 utilities may be said to accomplish certain services, they actually obtain services on behalf of a
 4673 user by means of requests to batch servers.

Table 3-1 Batch Utilities

<i>qalter</i>	<i>qmove</i>	<i>qrls</i>	<i>qstat</i>
<i>qdel</i>	<i>qmsg</i>	<i>qselect</i>	<i>qsub</i>
<i>qhold</i>	<i>qrerun</i>	<i>qsig</i>	

Client-server interaction takes place by means of the batch requests defined in this chapter. Because direct access to batch jobs and queues is limited to batch servers, clients and servers of different implementations can interoperate, since dependencies on private structures for batch jobs and queues are limited to batch servers. Also, batch servers may be clients of other batch servers.

3.1.2 Batch Queues

Two types of batch queue are described: *routing queues* and *execution queues*. When a batch job is placed in a routing queue, it is a candidate for routing. A batch job is removed from routing queues under the following conditions:

- The batch job has been routed to another queue.
- The batch job has been deleted from the batch queue.
- The batch job has been aborted.

When a batch job is placed in an execution queue, it is a candidate for execution.

A batch job is removed from an execution queue under the following conditions:

- The batch job has been executed and exited.
- The batch job has been aborted.
- The batch job has been deleted from the batch queue.
- The batch job has been moved to another queue.

Access to a batch queue is limited to the batch server that manages the batch queue. Clients never access a batch queue or a batch job directly, either to read or write information; all client access to batch queues or jobs takes place through batch servers.

3.1.3 Batch Job Creation

When a batch server creates a batch job on behalf of a client, it assigns a batch job identifier to the job. A batch job identifier consists of both a sequence number that is unique among the sequence numbers issued by that server and the name of the server. Since the batch server name is unique within a name space, the job identifier is likewise unique within the name space.

The batch server that creates a batch job returns the batch server-assigned job identifier to the client that requested the job creation. If the batch server routes or moves the job to another server, it sends the job identifier with the job. Once assigned, the job identifier of a batch job never changes.

4708 3.1.4 Batch Job Tracking

4709 Since a batch job may be moved after creation, the batch server name component of the job
 4710 identifier does not always indicate the location of the job. An implementation may provide a
 4711 batch job tracking mechanism, in which case the user generally does not need to know the
 4712 location of the job. However, an implementation is not required to provide a batch job tracking
 4713 mechanism, in which case the user must find routed jobs by probing the possible destinations.

4714 3.1.5 Batch Job Routing

4715 To route a batch job, a batch server either moves the job to some other queue that is managed by
 4716 the batch server, or requests that some other batch server accept the job.

4717 Each routing queue has one or more queues to which it can route batch jobs. The batch server
 4718 administrator creates routing queues.

4719 A batch server may route a batch job from a routing queue to another routing queue. Batch
 4720 servers shall prevent or otherwise handle cases of circular routing paths. As a deferred service, a
 4721 batch server routes jobs from the routing queues that it manages. The algorithm by which a
 4722 batch server selects a batch queue to which to route a batch job is implementation-dependent.

4723 A batch job need not be eligible for routing to all the batch queues fed by the routing queue from
 4724 which it is routed. A batch server that has been asked to accept the job may reject the request if
 4725 the job requires resources that are unavailable to that batch server, or if the client is not
 4726 authorized to access the batch server.

4727 Batch servers may route high-priority jobs before low-priority jobs, but, on other than
 4728 overloaded systems, the effect may be imperceptible to the user. If all the batch servers fed by a
 4729 routing queue reject requests to accept the job for reasons that are permanent, the batch server
 4730 that manages the job aborts the job. If all or some rejections are temporary, the batch server
 4731 should try to route the job again at some later point.

4732 The conformance document for an implementation shall list the reasons for rejecting the routing
 4733 of a batch job. The conformance document shall indicate the reasons for which the routing
 4734 should be retried later and the reasons for which the job should be aborted.

4735 3.1.6 Batch Job Execution

4736 To execute a batch job is to create a session leader (a process) that runs the shell program
 4737 indicated by the *Shell_Path* attribute of the job. The script is passed to the program as its
 4738 standard input. An implementation of the batch server may pass the script to the program by
 4739 other means. The implementation shall document the alternate means in the conformance
 4740 document. At the time a batch job begins execution, it is defined to enter the RUNNING state.
 4741 The primary program that is executed by a batch job is typically, though not necessarily, a shell
 4742 program.

4743 A batch server executes eligible jobs as a deferred service—no client request is necessary once
 4744 the batch job is created and eligible. However, the attributes of a batch job, such as the job hold
 4745 type, may render the job ineligible. A batch server scans the execution queues that it manages for
 4746 jobs that are eligible for execution. The algorithm by which the batch server selects eligible jobs
 4747 for execution is implementation-dependent.

4748 As part of creating the process for the batch job, the batch server opens the standard output and
 4749 standard error streams of the session.

4750 The attributes of a batch job may indicate that the batch server that executes the job is to send
 4751 mail to a list of users at the time it begins execution of the job.

4752 **3.1.7 Batch Job Exit**

4753 When the session leader of an executing job terminates, the job exits. As part of exiting a batch
 4754 job, the batch server that manages the job shall remove the job from the batch queue in which it
 4755 resides. The server shall transfer output files of the job to a location described by the attributes of
 4756 the job.

4757 The attributes of a batch job may indicate that the batch server that manages the job should send
 4758 mail to a list of users at the time the job exits.

4759 **3.1.8 Batch Job Abort**

4760 A batch server aborts jobs for which a required deferred service cannot be performed. The
 4761 attributes of a batch job may indicate that the batch server that aborts the job shall send mail to a
 4762 list of users at the time it aborts the job.

4763 **3.1.9 Batch Job States**

4764 A batch job is always in one of several states: QUEUED, RUNNING, HELD, WAITING,
 4765 EXITING, or TRANSITING. The state of a batch job determines the types of requests that the
 4766 batch server that manages the job can accept for the job. A batch server changes the state of a
 4767 batch job either in response to service requests from clients or as a result of deferred services
 4768 such as job execution or job routing.

4769 A batch job that is in the QUEUED state resides in a batch queue, but is still pending either
 4770 execution or routing, depending on the batch queue type.

4771 A batch job that resides in an execution queue and is executing is defined to be in the RUNNING
 4772 state. While a batch job is in the RUNNING state, a session leader is associated with the job.

4773 A batch job that resides in an execution queue, but is ineligible to run because of a hold attribute,
 4774 is defined to be in the HELD state. A batch job that is not held, but which must wait until a
 4775 future date and time before executing, is defined to be in the WAITING state.

4776 A batch job for which the session leader has terminated is defined to be in the EXITING state,
 4777 and the batch server that manages such a batch job cannot accept job modification requests that
 4778 affect the job. While a batch job is in the EXITING state, the batch server that manages the job is
 4779 staging output files and notifying clients of job completion. Once a batch job has exited, it no
 4780 longer exists as an object managed by a batch server.

4781 A batch job that is being moved from a routing queue to another queue is defined to be in the
 4782 TRANSITING state.

4783 **3.1.10 Batch Authorization**

4784 In order to access batch services, a user must have execute access to a batch client. For example,
 4785 to use the command language interface defined in this section, the user must be able to execute
 4786 the programs that embody those utilities.

4787 Clients, such as the utilities in this section, access batch services by means of requests to one or
 4788 more batch servers. To acquire the services of any given batch server, the user identifier under
 4789 which the client runs must be authorized to use that batch server.

4790 The user with an associated user name that creates a batch job owns the job and can perform
 4791 actions such as read, modify, delete, and move.

4792 A user identifier of the same value at a different host need not be the same user. For example,
 4793 user name *smith* at host **alpha** may or may not represent the same person as user name *smith* at

4794 host **beta**. Likewise, the same person may have access to different user names on different hosts.
 4795 An implementation may optionally provide an authorization mechanism that permits one user
 4796 name to access jobs under another user name.

4797 A process on a client host may be authorized to run processes under multiple user names at a
 4798 batch server host. Where appropriate, the utilities defined in this volume of
 4799 IEEE Std. 1003.1-200x provide a means for a user to choose from among such user names when
 4800 creating or modifying a batch job.

4801 **3.1.11 Batch Administration**

4802 The processing of a batch job by a batch server is affected by the attributes of the job. The
 4803 processing of a batch job may also be affected by the attributes of the batch queue in which the
 4804 job resides and by the status of the batch server that manages the job.

4805 A batch administrator is a user that is authorized to modify all the attributes of queues and jobs
 4806 and to change the status of a batch server. A batch operator is a user that is authorized to modify
 4807 some, but not all, of the attributes of jobs and queues, and may change the status of the batch
 4808 server.

4809 **3.1.12 Batch Notification**

4810 Whereas batch servers are persistent entities, clients are often transient. For example, the *qsub*
 4811 utility creates a batch job and exits. For this reason, batch servers notify users of batch job events
 4812 by sending mail to the user that owns the job, or to other designated users.

4813 **3.2 Batch Services**

4814 The presence of Batch Environment option services is indicated by the configuration variable
 4815 POSIX2_PBS. A conforming batch server provides services as defined in this section.

4816 A batch server provides batch services in two ways:

- 4817 1. The batch server provides a service at the request of a client.
- 4818 2. The batch server provides a deferred service as a result of a change in conditions
 4819 monitored by the batch server.

4820 If a batch server cannot complete a request, it rejects the request. If a batch server cannot
 4821 complete a deferred service for a batch job, the batch server aborts the batch job. Table 3-2 is a
 4822 summary of environment variables that shall be supported by an implementation of the batch
 4823 server and utilities.

4824 **Table 3-2** Environment Variable Summary

Variable	Description
PBS_DPREFIX	Defines the directive prefix (see <i>qsub</i>)
PBS_ENVIRONMENT	Batch Job is batch or interactive (see Section 3.2.2.1 on page 139)
PBS_JOBID	Job_identifier attribute of job (see Section 3.2.3.8 on page 151)
PBS_JOBNAME	Job_name attribute of job (see Section 3.2.3.8 on page 151)
PBS_O_HOME	Defines the <i>HOME</i> of the batch client (see <i>qsub</i>)
PBS_O_HOST	Defines the host name of the batch client (see <i>qsub</i>)
PBS_O_LANG	Defines the <i>LANG</i> of the batch client (see <i>qsub</i>)
PBS_O_LOGNAME	Defines the <i>LOGNAME</i> of the batch client (see <i>qsub</i>)
PBS_O_MAIL	Defines the <i>MAIL</i> of the batch client (see <i>qsub</i>)
PBS_O_PATH	Defines the <i>PATH</i> of the batch client (see <i>qsub</i>)
PBS_O_QUEUE	Defines the submit queue of the batch client (see <i>qsub</i>)
PBS_O_SHELL	Defines the <i>SHELL</i> of the batch client (see <i>qsub</i>)
PBS_O_TZ	Defines the <i>TZ</i> of the batch client (see <i>qsub</i>)
PBS_O_WORKDIR	Defines the working directory of the batch client (see <i>qsub</i>)
PBS_QUEUE	Defines the initial execution queue (see Section 3.2.2.1 on page 139)

3.2.1 Batch Job States

A batch job is always in one of several states: QUEUED, RUNNING, HELD, WAITING, EXITING, or TRANSITING. The state of a batch job determines the types of requests that the batch server that manages the batch job can accept for the batch job. A batch server changes the state of a batch job either in response to service requests from clients or as a result of deferred services, such as job execution or job routing.

A batch job that is in the QUEUED state resides in a queue but is still pending either execution or routing, depending on the queue type.

A batch server that queues a batch job in a routing queue shall put the batch job in the QUEUED state. A batch server that puts a batch job in an execution queue, but has not yet executed the batch job, shall put the batch job in the QUEUED state. A batch job that resides in an execution queue and is executing is defined to be in the RUNNING state. While a batch job is in the RUNNING state, a session leader is associated with the batch job.

A batch job that resides in an execution queue, but is ineligible to run because of a hold attribute, is defined to be in the HELD state.

A batch job that is not held, but must wait until a future date and time before executing, is defined to be in the WAITING state.

When the session leader associated with a running job exits, the batch job shall be placed in the EXITING state.

A batch job for which the session leader has terminated is defined to be in the EXITING state, and the batch server that manages such a batch job cannot accept job modification requests that affect the batch job. While a batch job is in the EXITING state, the batch server that manages the batch job is staging output files and notifying clients of job completion. Once a batch job has exited, it no longer exists as an object managed by a batch server.

A batch job that is being moved from a routing queue to another queue is defined to be in the TRANSITING state.

4869 When a batch job in a routing queue has been selected to be moved to a new destination, then
 4870 the batch job is in either the QUEUED state or the TRANSITING state, depending on the batch
 4871 server implementation.

4872 Batch jobs with either a *Execution_Time* attribute value set in the future or a *Hold_Types* attribute
 4873 of value not equal to NO_HOLD, or both, may be routed or held in the routing queue. An
 4874 implementation shall document the treatment of jobs with the *Execution_Time* or *Hold_Types*
 4875 attributes in a routing queue.

4876 When a batch job in a routing queue has not been selected to be moved to a new destination and
 4877 the batch job has a *Hold_Types* attribute value of other than NO_HOLD, then the job should be in
 4878 the HELD state.

4879 **Note:** The effect of a hold upon a batch job in a routing queue is implementation-
 4880 dependent. The implementation should use the state that matches whether the batch
 4881 job can route with a hold or not.

4882 When a batch job in a routing queue has not been selected to be moved to a new destination and
 4883 the batch job has:

- 4884 • A *Hold_Types* attribute value of NO_HOLD
- 4885 • An *Execution_Time* attribute in the past

4886 then the batch job shall be in the QUEUED state.

4887 When a batch job in a routing queue has not been selected to be moved to a new destination and
 4888 the batch job has:

- 4889 • A *Hold_Types* attribute value of NO_HOLD
- 4890 • A *Execution_Time* attribute in the future

4891 then the batch job may be in the WAITING state.

4892 **Note:** The effect of a future execution time upon a batch job in a routing queue is
 4893 implementation-dependent. The implementation should use the state that matches
 4894 whether the batch job can route with a hold or not.

4895 Table 3-3 on page 140 describes the next state of a batch job, given the current state of the batch
 4896 job and the type of request. Table 3-4 on page 141 describes the response of a batch server to a
 4897 request, given the current state of the batch job and the type of request.

4898 3.2.2 Deferred Batch Services

4899 This section describes the deferred services performed by batch servers: job execution, job
 4900 routing, job exit, job abort, and the rerunning of jobs after a restart.

4901 3.2.2.1 Batch Job Execution

4902 To execute a batch job is to create a session leader (a process) that runs the shell program
 4903 indicated by the *Shell_Path_List* attribute of the batch job. The script is passed to the program as
 4904 its standard input. An implementation of the batch server may pass the script to the program by
 4905 other means. The implementation shall document the alternate means in the conformance
 4906 document. At the time a batch job begins execution, it is defined to enter the RUNNING state.

Table 3-3 Next State Table

Request Type	Current State						
	X	Q	R	H	W	E	T
<i>Queue Batch Job Request</i>	Q	e	e	e	e	e	e
<i>Modify Batch Job Request</i>	e	Q	R	H	W	e	T
<i>Delete Batch Job Request</i>	e	X	E	X	X	E	X
<i>Batch Job Message Request</i>	e	Q	R	H	W	E	T
<i>Rerun Batch Job Request</i>	e	e	Q	e	e	e	e
<i>Signal Batch Job Request</i>	e	e	R	H	W	e	e
<i>Batch Job Status Request</i>	e	Q	R	H	W	E	T
<i>Batch Queue Status Request</i>	X	Q	R	H	W	E	T
<i>Server Status Request</i>	X	Q	R	H	W	E	T
<i>Select Batch Jobs Request</i>	X	Q	R	H	W	E	T
<i>Move Batch Job Request</i>	e	Q	R	H	W	e	T
<i>Hold Batch Job Request</i>	e	H	R/H	H	H	e	T
<i>Release Batch Job Request</i>	Q	R	Q/W/H	W	e	T	
<i>Server Shutdown Request</i>	X	Q	Q	H	W	E	T
<i>Locate Batch Job Request</i>	e	Q	R	H	W	E	T

Legend

X Nonexistent

Q QUEUED

R RUNNING

H HELD

W WAITING

E EXITING

T TRANSITING

e Error

A batch server that has an execution queue containing jobs is said to own the queue and manage the batch jobs in that queue. A batch server that has been started shall execute the batch jobs in the execution queues owned by the batch server. The batch server shall schedule for execution those jobs in the execution queues that are in the QUEUED state. The algorithm for scheduling jobs is implementation-dependent.

A batch server that executes a batch job shall create, in the environment of the session leader of the batch job, an environment variable named *PBS_ENVIRONMENT*, the value of which is the string *PBS_BATCH* encoded in the portable character set.

A batch server that executes a batch job shall create, in the environment of the session leader of the batch job, an environment variable named *PBS_QUEUE*, the value of which is the name of the execution queue of the batch job encoded in the portable character set.

To rerun a batch job is to requeue a batch job that is currently executing and then kill the session leader of the executing job by sending a SIGKILL prior to completion; see Section 3.2.3.11 on page 153. A batch server that reruns a batch job shall append the standard output and standard error files of the batch job to the corresponding files of the previous execution, if they exist, with appropriate annotation. If either file does not exist, that file shall be created as in normal execution.

Table 3-4 Results/Output Table

Request Type	Current State						
	X	Q	R	H	W	E	T
Queue Batch Job Request	O	e	e	e	e	e	e
Modify Batch Job Request	e	O	e	O	O	e	e
Delete Batch Job Request	e	O	O	O	O	e	O
Batch Job Message Request	e	e	O	e	e	e	e
Rerun Batch Job Request	e	e	O	e	e	e	e
Signal Batch Job Request	e	e	O	e	e	e	e
Batch Job Status Request	e	O	O	O	O	O	O
Batch Queue Status Request	O	O	O	O	O	O	O
Server Status Request	O	O	O	O	O	O	O
Select Batch Job Request	e	O	O	O	O	O	O
Move Batch Job Request	e	O	O	O	O	e	e
Hold Batch Job Request	e	O	O	O	O	e	e
Release Batch Job Request	e	O	e	O	O	e	e
Server Shutdown Request	O	O	e	O	O	e	e
Locate Batch Job Request	e	O	O	O	O	O	O

Legend

O OK

e Error message

The execution of a batch job by a batch server is controlled by job, queue, and server attributes, as defined in this section.

Account_Name Attribute

Batch accounting is an optional feature of batch servers. If a batch server implements accounting, the statements in this section apply and the configuration variable `POSIX2_PBS_ACCOUNTING` shall be set to 1.

A batch server that executes a batch job shall charge the account named in the *Account_Name* attribute of the batch job for resources consumed by the batch job.

If the *Account_Name* attribute of the batch job is absent from the batch job attribute list or is altered while the batch job is in execution, the batch server action is implementation-dependent.

Checkpoint Attribute

Batch checkpointing is an optional feature of batch servers. If a batch server implements checkpointing, the statements in this section apply and the configuration variable `POSIX2_PBS_CHECKPOINT` shall be set to 1.

There are two attributes associated with the checkpointing feature: *Checkpoint* and *Minimum_Cpu_Interval*. *Checkpoint* is a batch job attribute, while *Minimum_Cpu_Interval* is a queue attribute. An implementation that does not support checkpointing shall support the *Checkpoint* job attribute to the extent that the batch server shall maintain and pass this attribute to other servers.

The behavior of a batch server that executes a batch job for which the value of the *Checkpoint* attribute is `CHECKPOINT_UNSPECIFIED` is implementation-dependent. The implementation shall document the behavior of the batch server. A batch server that executes a batch job for

4994 which the value of the *Checkpoint* attribute is NO_CHECKPOINT shall not checkpoint the batch
4995 job.

4996 A batch server that executes a batch job for which the value of the *Checkpoint* attribute is
4997 CHECKPOINT_AT_SHUTDOWN shall checkpoint the batch job only when the batch server
4998 accepts a request to shut down during the time when the batch job is in the RUNNING state.

4999 A batch server that executes a batch job for which the value of the *Checkpoint* attribute is
5000 CHECKPOINT_AT_MIN_CPU_INTERVAL shall checkpoint the batch job at the interval
5001 specified by the *Minimum_Cpu_Interval* attribute of the queue for which the batch job has been
5002 selected. The *Minimum_Cpu_Interval* attribute shall be specified in units of CPU minutes.

5003 A batch server that executes a batch job for which the value of the *Checkpoint* attribute is an
5004 unsigned integer shall checkpoint the batch job at an interval that is the value of either the
5005 *Checkpoint* attribute, or the *Minimum_Cpu_Interval* attribute of the queue for which the batch job
5006 has been selected, whichever is greater. Both intervals shall be in units of CPU minutes. When
5007 the *Minimum_Cpu_Interval* attribute is greater than the *Checkpoint* attribute, the batch job shall
5008 write a warning message to the standard error stream of the batch job.

5009 **Error_Path Attribute**

5010 The *Error_Path* attribute of a running job cannot be changed by a *Modify Batch Job Request*. When
5011 the *Join_Path* attribute of the batch job is set to the value FALSE and the *Keep_Files* attribute of
5012 the batch job does not contain the value KEEP_STD_ERROR, a batch server that executes a batch
5013 job shall perform one of the following actions:

- 5014 • Set the standard error stream of the session leader of the batch job to the path described by
5015 the value of the *Error_Path* attribute of the batch job.
- 5016 • Buffer the standard error of the session leader of the batch job until completion of the batch
5017 job, and when the batch job exits return the contents to the destination described by the value
5018 of the *Error_Path* attribute of the batch job. Where the batch server buffers standard error is
5019 implementation-dependent.

5020 Applications shall not rely on having access to the standard error of a batch job prior to the
5021 completion of the batch job.

5022 When the *Error_Path* attribute does not specify a host name, then the batch server shall retain the
5023 standard error of the batch job on the host of execution.

5024 When the *Error_Path* attribute does specify a host name and the *Keep_Files* attribute does not
5025 contain the value KEEP_STD_ERROR, then the final destination of the standard error of the
5026 batch job shall be on the host whose host name is specified.

5027 If the path indicated by the value of the *Error_Path* attribute of the batch job is a relative path, the
5028 batch server shall expand the path relative to the home directory of the user on the host to which
5029 the file is being returned.

5030 When the batch server buffers the standard error of the batch job and the file cannot be opened
5031 for write upon completion of the batch job, then the server shall place the standard error in an
5032 implementation-dependent location and notify the user of the location via mail. It shall be
5033 possible for the user to process this mail using the *mailx* utility.

5034 If a batch server that does not buffer the standard error cannot open the standard error path of
5035 the batch job for write access, then the batch server shall abort the batch job.

5036 **Execution_Time Attribute**

5037 A batch server shall not execute a batch job before the time represented by the value of the
 5038 *Execution_Time* attribute of the batch job. The *Execution_Time* attribute is defined in seconds since
 5039 the Epoch.

5040 **Hold_Types Attribute**

5041 A batch server shall support the following hold types:

5042 **s** Can be set or released by a user with at least a privilege level of batch administrator
 5043 (SYSTEM).

5044 **o** Can be set or released by a user with at least a privilege level of batch operator
 5045 (OPERATOR).

5046 **u** Can be set or released by the user with at least a privilege level of user, where the user is
 5047 defined in the *Job_Owner* attribute (USER).

5048 **n** Indicates that none of the *Hold_Types* attributes are set (NO_HOLD).

5049 An implementation may define other hold types. The conformance document for an
 5050 implementation shall describe any additional hold types, how they are specified, their internal
 5051 representation, their behavior, and how they affect the behavior of other utilities.

5052 The value of the *Hold_Types* attribute shall be the union of the valid hold types (**ss**, **oo**, **uu**, and
 5053 any implementation-dependent hold types), or **nn**.

5054 A batch server shall not execute a batch job if the *Hold_Types* attribute of the batch job has a
 5055 value other than NO_HOLD. If the *Hold_Types* attribute of the batch job has a value other than
 5056 NO_HOLD, the batch job shall be in the HELD state.

5057 **Job_Owner Attribute**

5058 The *Job_Owner* attribute consists of a pair of user name and host name values of the form:

5059 `username@hostname`

5060 A batch server that accepts a *Queue Batch Job Request* shall set the *Job_Owner* attribute to a string
 5061 that is the *username@hostname* of the user who submitted the job.

5062 **Join_Path Attribute**

5063 A batch server that executes a batch job for which the value of the *Join_Path* attribute is TRUE
 5064 shall ignore the value of the *Error_Path* attribute and merge the standard error of the batch job
 5065 with the standard output of the batch job.

5066 **Keep_Files Attribute**

5067 A batch server that executes a batch job for which the value of the *Keep_Files* attribute includes
 5068 the value KEEP_STD_OUTPUT shall retain the standard output of the batch job on the host
 5069 where execution occurs. The standard output shall be retained in the home directory of the user
 5070 under whose user ID the batch job is executed and the file name shall be the default file name for
 5071 the standard output as defined under the **-o** option of the *qsub* utility. The *Output_Path* attribute
 5072 is not modified.

5073 A batch server that executes a batch job for which the value of the *Keep_Files* attribute includes
 5074 the value KEEP_STD_ERROR shall retain the standard error of the batch job on the host where
 5075 execution occurs. The standard error shall be retained in the home directory of the user under
 5076 whose user ID the batch job is executed and the file name shall be the default file name for

5077 standard error as defined under the `-e` option of the *qsub* utility. The *Error_Path* attribute is not
 5078 modified.

5079 A batch server that executes a batch job for which the value of the *Keep_Files* attribute includes
 5080 values other than `KEEP_STD_OUTPUT` and `KEEP_STD_ERROR` shall retain these other files on
 5081 the host where execution occurs. These files shall be retained in the home directory of the user
 5082 under whose user identifier the batch job is executed and the file names shall be the default file
 5083 names for the files as defined in the conformance document for the implementation.

5084 **Mail_Points and Mail_Users Attributes**

5085 A batch server that executes a batch job for which one of the values of the *Mail_Points* attribute is
 5086 the value `MAIL_AT_BEGINNING` shall send a mail message to each user account listed in the
 5087 *Mail_Users* attribute of the batch job.

5088 The mail message shall contain at least the batch job identifier, queue, and server at which the
 5089 batch job currently resides, and the *Job_Owner* attribute.

5090 **Output_Path Attribute**

5091 The *Output_Path* attribute of a running job cannot be changed by a *Modify Batch Job Request*.
 5092 When the *Keep_Files* attribute of the batch job does not contain the value `KEEP_STD_OUTPUT`, a
 5093 batch server that executes a batch job shall either:

5094 • Set the standard output stream of the session leader of the batch job to the destination
 5095 described by the value of the *Output_Path* attribute of the batch job.

5096 or:

5097 • Buffer the standard output of the session leader of the batch job until completion of the batch
 5098 job, and when the batch job exits return the contents to the destination described by the value
 5099 of the *Output_Path* attribute of the batch job.

5100 When the *Output_Path* attribute does not specify a host name, then the batch server shall retain
 5101 the standard output of the batch job on the host of execution.

5102 When the *Keep_Files* attribute does not contain the value `KEEP_STD_OUTPUT` and the
 5103 *Output_Path* attribute does specify a host name, then the final destination of the standard output
 5104 of the batch job shall be on the host specified.

5105 If the path specified in the *Output_Path* attribute of the batch job is a relative path, the batch
 5106 server shall expand the path relative to the home directory of the user on the host to which the
 5107 file is being returned.

5108 Whether or not the batch server buffers the standard output of the batch job until completion of
 5109 the batch job is implementation-dependent. Applications shall not rely on having access to the
 5110 standard output of a batch job prior to the completion of the batch job.

5111 When the batch server does buffer the standard output of the batch job and the file cannot be
 5112 opened for write upon completion of the batch job, then the batch server shall place the standard
 5113 output in an implementation-dependent location and notify the user of the location via mail. It
 5114 shall be possible for the user to process this mail using the *mailx* utility.

5115 If a batch server that does not buffer the standard output cannot open the standard output path
 5116 of the batch job for write access, then the batch server shall abort the batch job.

5117 **Priority Attribute**

5118 A batch server implementation might choose to preferentially execute a batch job based on the
 5119 *Priority* attribute. The interpretation of the batch job *Priority* attribute by a batch server is
 5120 implementation-dependent. If an implementation uses the *Priority* attribute, it shall interpret
 5121 larger values of the *Priority* attribute to mean the batch job shall be preferentially selected for
 5122 execution.

5123 **Rerunable Attribute**

5124 A batch job that began execution but did not complete, because the batch server either shut
 5125 down or terminated abnormally, shall be requeued if the *Rerunable* attribute of the batch job has
 5126 the value TRUE.

5127 If a batch job, which was requeued after beginning execution but prior to completion, has a valid
 5128 checkpoint file and the batch server supports checkpointing, then the batch job shall be restarted
 5129 from the last valid checkpoint.

5130 If the batch job cannot be restarted from a checkpoint, then when a batch job has a *Rerunable*
 5131 attribute value of TRUE and was requeued after beginning execution but prior to completion,
 5132 the batch server shall place the batch job into execution at the beginning of the job.

5133 When a batch job has a *Rerunable* attribute value other than TRUE and was requeued after
 5134 beginning execution but prior to completion, and the batch job cannot be restarted from a
 5135 checkpoint, then the batch server shall abort the batch job.

5136 **Resource_List Attribute**

5137 A batch server that executes a batch job shall establish the resource limits of the session leader of
 5138 the batch job according to the values of the *Resource_List* attribute of the batch job. Resource
 5139 limits shall be enforced by an implementation-dependent method.

5140 **Shell_Path_List Attribute**

5141 The *Shell_Path_List* job attribute consists of a list of pairs of path name and host name values.
 5142 The host name component can be omitted, in which case the path name serves as the default
 5143 path name when a batch server cannot find the name of the host on which it is running in the
 5144 list.

5145 A batch server that executes a batch job shall select, from the value of the *Shell_Path_List*
 5146 attribute of the batch job, a path name where the shell to execute the batch job shall be found.
 5147 The batch server shall select the path name, in order of preference, according to the following
 5148 methods:

- 5149 • Select the path name that contains the name of the host on which the batch server is running.
- 5150 • Select the path name for which the host name has been omitted.
- 5151 • Select the path name for the login shell of the user under which the batch job is to execute.

5152 If the shell path value selected is an invalid path name, the batch server shall abort the batch job.

5153 If the value of the selected path name from the *Shell_Path_List* attribute of the batch job
 5154 represents a partial path, the batch server shall expand the path relative to a path that is
 5155 implementation-dependent.

5156 The batch server that executes the batch job shall execute the program that was selected from the
 5157 *Shell_Path_List* attribute of the batch job. The batch server shall pass the path to the script of the
 5158 batch job as the first argument to the shell program.

5159 **User_List Attribute**

5160 The *User_List* job attribute consists of a list of pairs of user name and host name values. The host
 5161 name component can be omitted, in which case the user name serves as a default when a batch
 5162 server cannot find the name of the host on which it is running in the list.

5163 A batch server that executes a batch job shall select, from the value of the *User_List* attribute of
 5164 the batch job, a user name under which to create the session leader. The server shall select the
 5165 user name, in order of preference, according to the following methods:

- 5166 • Select the user name of a value that contains the name of the host on which the batch server
 5167 executes.
- 5168 • Select the user name of a value for which the host name has been omitted.
- 5169 • Select the user name from the *Job_Owner* attribute of the batch job.

5170 **Variable_List Attribute**

5171 A batch server that executes a batch job shall create, in the environment of the session leader of
 5172 the batch job, each environment variable listed in the *Variable_List* attribute of the batch job, and
 5173 set the value of each such environment variable to that of the corresponding variable in the
 5174 variable list.

5175 3.2.2.2 **Batch Job Routing**

5176 To route a batch job is to select a queue from a list and move the batch job to that queue.

5177 A batch server that has routing queues, which have been started, shall route the jobs in the
 5178 routing queues owned by the batch server. A batch server is allowed to delay the routing of a
 5179 batch job. The algorithm for selecting a batch job and the queue to which it will be routed is
 5180 implementation-dependent.

5181 When a routing queue has multiple possible destinations specified, then the precedence of the
 5182 destination is implementation-dependent.

5183 A batch server that routes a batch job to a queue at another server shall move the batch job into
 5184 the target queue with a *Queue Batch Job Request*.

5185 If the target server rejects the *Queue Batch Job Request*, the routing server shall retry routing the
 5186 batch job or abort the batch job. A batch server that retries failed routings shall provide a means
 5187 for the batch administrator to specify the number of retries and the minimum period of time
 5188 between retries. The means by which an administrator specifies the number of retries and the
 5189 delay between retries is implementation-dependent. When the number of retries specified by the
 5190 batch administrator has been exhausted, the batch server shall abort the batch job and perform
 5191 the functions of *Batch Job Exit*; see Section 3.2.2.3.

5192 3.2.2.3 **Batch Job Exit**

5193 For each job in the EXITING state, the batch server that exited the batch job shall perform the
 5194 following deferred services in the order specified:

- 5195 1. If buffering standard error, move that file into the location specified by the *Error_Path*
 5196 attribute of the batch job.
- 5197 2. If buffering standard output, move that file into the location specified by the *Output_Path*
 5198 attribute of the batch job.
- 5199 3. If the *Mail_Points* attribute of the batch job includes MAIL_AT_EXIT, send mail to the users
 5200 listed in the *Mail_Users* attribute of the batch job. The mail message shall contain at least

5201 the batch job identifier, queue, and server at which the batch job currently resides, and the
5202 *Job_Owner* attribute.

5203 4. Remove the batch job from the queue.

5204 If a batch server that buffers the standard error output cannot return the standard error file to
5205 the standard error path at the time the batch job exits, the batch server shall do one of the
5206 following:

- 5207 • Mail the standard error file to the batch job owner.
- 5208 • Save the standard error file and mail the location and name of the file where the standard
5209 error is stored to the batch job owner.
- 5210 • Save the standard error file and notify the user by other means, in which case the
5211 conformance document for the implementation shall document the method of notification.

5212 If a batch server that buffers the standard output cannot return the standard output file to the
5213 standard output path at the time the batch job exits, the batch server shall do one of the
5214 following:

- 5215 • Mail the standard output file to the batch job owner.
- 5216 • Save the standard output file and mail the location and name of the file where the standard
5217 output is stored to the batch job owner.
- 5218 • Save the standard output file and notify the user by other means, in which case the
5219 conformance document for the implementation shall document the method of notification.

5220 At the conclusion of job exit processing, the batch job is no longer managed by a batch server.

5221 3.2.2.4 *Batch Server Restart*

5222 A batch server that has been either shutdown or terminated abnormally, and has returned to
5223 operation, is said to have *restarted*.

5224 Upon restarting, a batch server shall requeue those jobs managed by the batch server that were
5225 in the RUNNING state at the time the batch server shut down and for which the *Rerunable*
5226 attribute of the batch job has the value TRUE.

5227 Queues are defined to be non-volatile. A batch server shall store the content of queues that it
5228 controls in such a way that server and system shutdowns do not erase the content of the queues.

5229 3.2.2.5 *Batch Job Abort*

5230 A batch server that cannot perform a deferred service for a batch job shall abort the batch job.

5231 A batch server that aborts a batch job shall perform the following services:

- 5232 • Delete the batch job from the queue in which it resides.
- 5233 • If the *Mail_Points* attribute of the batch job includes the value MAIL_AT_ABORT, send mail
5234 to the users listed in the value of the *Mail_Users* attribute of the job. The mail message shall
5235 contain at least the batch job identifier, queue, and server at which the batch job currently
5236 resides, the *Job_Owner* attribute, and the reason for the abort.
- 5237 • If the batch job was in the RUNNING state, terminate the session leader of the executing job
5238 by sending the session leader a SIGKILL, place the batch job in the EXITING state, and
5239 perform the services of *Batch Job Exit*.

5240 3.2.3 Requested Batch Services

5241 This section describes the services provided by batch servers in response to requests from
 5242 clients. Table 3-5 summarizes the current set of batch service requests and for each gives its type
 5243 (deferred or not) and whether it is an optional function.

5244 If a request is rejected because the batch client is not authorized to perform the action, the batch
 5245 server shall return the same status as when the batch job does not exist.

5246 3.2.3.1 Delete Batch Job Request

5247 A batch job is defined to have been deleted when it has been removed from the queue in which it
 5248 resides and not instantiated in another queue. A client requests that the server that manages a
 5249 batch job delete the batch job. Such a request is called a *Delete Batch Job Request*.

5250 **Table 3-5** Batch Services Summary

Batch Service	Deferred	Optional
<i>Batch Job Execution</i>	Yes	No
<i>Batch Job Routing</i>	Yes	No
<i>Batch Job Exit</i>	Yes	No
<i>Batch Server Restart</i>	Yes	No
<i>Batch Job Abort</i>	Yes	No
<i>Queue Batch Job Request</i>	No	No
<i>Modify Batch Job Request</i>	No	No
<i>Delete Batch Job Request</i>	No	No
<i>Batch Job Message Request</i>	No	Yes
<i>Rerun Batch Job Request</i>	No	No
<i>Signal Batch Job Request</i>	No	No
<i>Batch Job Status Request</i>	No	No
<i>Batch Queue Status Request</i>	No	No
<i>Server Status Request</i>	No	No
<i>Select Batch Jobs Request</i>	No	No
<i>Move Batch Job Request</i>	No	No
<i>Hold Batch Job Request</i>	No	No
<i>Release Batch Job Request</i>	No	No
<i>Server Shutdown Request</i>	No	No
<i>Locate Batch Job Request</i>	No	Yes
<i>Track Batch Job Request</i>	No	Yes

5273 A batch server shall reject a *Delete Batch Job Request* if any of the following statements are true:

- 5274 • The user of the batch client is not authorized to delete the designated job.
- 5275 • The designated job is not managed by the batch server.
- 5276 • The designated job is in a state inconsistent with the delete request.

5277 A batch server may reject a *Delete Batch Job Request* for other reasons. The conformance document
 5278 for an implementation shall describe the reasons for which a *Delete Batch Job Request* may be
 5279 rejected. The conformance document for an implementation shall describe the method used to
 5280 determine whether the user of a client is authorized to perform the requested action.

5281 A batch server requested to delete a batch job shall delete the batch job if the batch job exists and
 5282 is not in the EXITING state.

5283 A batch server that deletes a batch job in the RUNNING state shall send a SIGKILL signal to the
 5284 session leader of the batch job. A batch server may send additional signals to the session leader
 5285 of the job prior to sending the SIGKILL signal. The conformance document for such a batch
 5286 server shall document the signals that are sent to the session leader.

5287 A batch server that deletes a batch job in the RUNNING state shall place the batch job in the
 5288 EXITING state after it has killed the session leader of the batch job and shall perform the services
 5289 of batch job exit.

5290 3.2.3.2 *Hold Batch Job Request*

5291 A batch client can request that the batch server add one or more holds to a batch job. Such a
 5292 request is called a *Hold Batch Job Request*.

5293 A batch server shall reject a *Hold Batch Job Request* if any of the following statements are true:

- 5294 • The batch server does not support one or more of the requested holds to be added to the
 5295 batch job.
- 5296 • The user of the batch client is not authorized to add one or more of the requested holds to the
 5297 batch job.
- 5298 • The batch server does not manage the specified job.
- 5299 • The designated job is in the EXITING state.

5300 A batch server may reject a *Hold Batch Job Request* for other reasons. The conformance document
 5301 for an implementation shall document the reasons for which a *Hold Batch Job Request* may be
 5302 rejected. The conformance document for an implementation shall describe the method used to
 5303 determine whether the user of a client is authorized to perform the requested action.

5304 A batch server that accepts a *Hold Batch Job Request* for a batch job in the RUNNING state shall
 5305 place a hold on the batch job. The conformance document shall describe what effect, if any, the
 5306 hold will have on a batch job in the RUNNING state.

5307 A batch server that accepts a *Hold Batch Job Request* shall add each type of hold listed in the *Hold*
 5308 *Batch Job Request*, that is not already present, to the value of the *Hold_Types* attribute of the batch
 5309 job.

5310 3.2.3.3 *Batch Job Message Request*

5311 *Batch Job Message Request* is an optional feature of batch servers. If an implementation supports
 5312 *Batch Job Message Request*, the statements in this section apply and the configuration variable
 5313 POSIX2_PBS_MESSAGE shall be set to 1.

5314 A batch client can request that a batch server write a message into certain output files of a batch
 5315 job. Such a request is called a *Batch Job Message Request*.

5316 A batch server shall reject a *Batch Job Message Request* if any of the following statements are true:

- 5317 • The batch server does not support sending messages to jobs.
- 5318 • The user of the batch client is not authorized to post a message to the designated job.
- 5319 • The designated job does not exist on the batch server.
- 5320 • The designated job is not in the RUNNING state.

5321 A batch server may reject a *Batch Job Message Request* for other reasons. The conformance
 5322 document for an implementation shall describe the reasons for which a *Batch Job Message Request*
 5323 may be rejected. The conformance document for an implementation shall describe the method

5324	used to determine whether the user of a client is authorized to perform the requested action.	
5325	A batch server that accepts a <i>Batch Job Message Request</i> shall write the message sent by the batch	
5326	client into the files indicated by the batch client.	
5327	3.2.3.4 Batch Job Status Request	
5328	A batch client can request that a batch server respond with the status and attributes of a batch	
5329	job. Such a request is called a <i>Batch Job Status Request</i> .	
5330	A batch server shall reject a <i>Batch Job Status Request</i> if any of the following statements are true:	
5331	• The user of the batch client is not authorized to query the status of the designated job.	
5332	• The designated job is not managed by the batch server.	
5333	A batch server may reject a <i>Batch Job Status Request</i> for other reasons. The conformance	
5334	document for an implementation shall describe the reasons for which a <i>Batch Job Status Request</i>	
5335	may be rejected. The conformance document for an implementation shall describe the method	
5336	used to determine whether the user of a client is authorized to perform the requested action.	
5337	A batch server that accepts a <i>Batch Job Status Request</i> shall return a <i>Batch Job Status Message</i> to the	
5338	batch client.	
5339	A batch server may return other information in response to a <i>Batch Job Status Request</i> .	
5340	3.2.3.5 Locate Batch Job Request	
5341	<i>Locate Batch Job Request</i> is an optional feature of batch servers. If an implementation supports	
5342	<i>Locate Batch Job Request</i> , the statements in this section apply and the configuration variable	
5343	POSIX2_PBS_LOCATE shall be set to 1.	
5344	A batch client can ask a batch server to respond with the location of a batch job that was created	
5345	by the batch server. Such a request is called a <i>Locate Batch Job Request</i> .	
5346	A batch server that accepts a <i>Locate Batch Job Request</i> shall return a <i>Batch Job Location Message</i> to	
5347	the batch client.	
5348	A batch server may reject a <i>Locate Batch Job Request</i> for a batch job that was not created by that	
5349	server.	
5350	A batch server may reject a <i>Locate Batch Job Request</i> for a batch job that is no longer managed by	
5351	that server; that is, for a batch job that is not in a queue owned by that server.	
5352	A batch server may reject a <i>Locate Batch Job Request</i> for other reasons. The conformance	
5353	document for an implementation shall document the reasons for which a <i>Locate Batch Job Request</i>	
5354	may be rejected.	
5355	3.2.3.6 Modify Batch Job Request	
5356	Batch clients modify (alter) the attributes of a batch job by making a request to the server that	
5357	manages the batch job. Such a request is called a <i>Modify Batch Job Request</i> .	
5358	A batch server shall reject a <i>Modify Batch Job Request</i> if any of the following statements are true:	
5359	• The user of the batch client is not authorized to make the requested modification to the batch	
5360	job.	
5361	• The designated job is not managed by the batch server.	
5362	• The requested modification is inconsistent with the state of the batch job.	

5363	<ul style="list-style-type: none"> • An unrecognized resource is requested for a batch job in an execution queue. 	
5364	A batch server may reject a <i>Modify Batch Job Request</i> for other reasons. The conformance	
5365	document for an implementation shall describe the reasons for which a <i>Modify Batch Job Request</i>	
5366	may be rejected. The conformance document for an implementation shall describe the method	
5367	used to determine whether the user of a client is authorized to perform the requested action.	
5368	A batch server that accepts a <i>Modify Batch Job Request</i> shall modify all the specified attributes of	
5369	the batch job. A batch server that rejects a <i>Modify Batch Job Request</i> shall modify none of the	
5370	attributes of the batch job.	
5371	If the servicing by a batch server of an otherwise valid request would result in no change, then	
5372	the batch server shall indicate successful completion of the request.	
5373	3.2.3.7 <i>Move Batch Job Request</i>	
5374	A batch client can request that a batch server move a batch job to another destination. Such a	
5375	request is called a <i>Move Batch Job Request</i> .	
5376	A batch server shall reject a <i>Move Batch Job Request</i> if any of the following statements are true:	
5377	<ul style="list-style-type: none"> • The user of the batch client is not authorized to remove the designated job from the queue in 	
5378	which the batch job resides.	
5379	<ul style="list-style-type: none"> • The user of the batch client is not authorized to move the designated job to the destination. 	
5380	<ul style="list-style-type: none"> • The designated job is not managed by the batch server. 	
5381	<ul style="list-style-type: none"> • The designated job is in the EXITING state. 	
5382	<ul style="list-style-type: none"> • The destination is inaccessible. 	
5383	A batch server can reject a <i>Move Batch Job Request</i> for other reasons. The conformance document	
5384	for an implementation shall describe the reasons for which a <i>Move Batch Job Request</i> may be	
5385	rejected. The conformance document for an implementation shall describe the method used to	
5386	determine whether the user of a client is authorized to perform the requested action.	
5387	A batch server that accepts a <i>Move Batch Job Request</i> shall perform the following services:	
5388	<ul style="list-style-type: none"> • Queue the designated job at the destination. 	
5389	<ul style="list-style-type: none"> • Remove the designated job from the queue in which the batch job resides. 	
5390	If the destination resides on another batch server, the batch server shall queue the batch job at	
5391	the destination by sending a <i>Queue Batch Job Request</i> to the other server. If the <i>Queue Batch Job</i>	
5392	<i>Request</i> fails, the batch server shall reject the <i>Move Batch Job Request</i> . If the <i>Queue Batch Job Request</i>	
5393	succeeds, the batch server shall remove the batch job from its queue.	
5394	The batch server shall not modify any attributes of the batch job.	
5395	3.2.3.8 <i>Queue Batch Job Request</i>	
5396	A batch queue is controlled by one and only one batch server. A batch server is said to own the	
5397	queues that it controls. Batch clients make requests of batch servers to have jobs queued. Such a	
5398	request is called a <i>Queue Batch Job Request</i> .	
5399	A batch server requested to queue a batch job for which the queue is unspecified shall select a	
5400	queue for the batch job. Such a queue is called the <i>default queue</i> of the batch server. The	
5401	conformance document for the implementation shall document the means by which the batch	
5402	server determines the default queue. The implementation shall provide the means for a batch	
5403	administrator to specify the default queue. The queue, whether specified or defaulted, is called	

5404	the <i>target queue</i> .	
5405	A batch server shall reject a <i>Queue Batch Job Request</i> if any of the following statements are true:	
5406	<ul style="list-style-type: none"> • The client is not authorized to create a batch job in the target queue. 	
5407	<ul style="list-style-type: none"> • The request specifies a queue that does not exist on the batch server. 	
5408	<ul style="list-style-type: none"> • The target queue is an execution queue and the batch server cannot satisfy a resource requirement of the batch job. 	
5409		
5410	<ul style="list-style-type: none"> • The target queue is an execution queue and an unrecognized resource is requested. 	
5411	<ul style="list-style-type: none"> • The target queue is an execution queue, the batch server does not support checkpointing, and the value of the <i>Checkpoint</i> attribute of the batch job is not NO_CHECKPOINT. 	
5412		
5413	<ul style="list-style-type: none"> • The job requires access to a user identifier that the batch client is not authorized to access. 	
5414	A batch server may reject a <i>Queue Batch Job Request</i> for other reasons. The conformance document for an implementation shall document the reasons for which a <i>Queue Batch Job Request</i>	
5415	may be rejected.	
5416		
5417	A batch server that accepts a <i>Queue Batch Job Request</i> for a batch job for which the	
5418	PBS_O_QUEUE value is missing from the value of the <i>Variable_List</i> attribute of the batch job	
5419	shall add that variable to the list and set the value to the name of the target queue. Once set, no	
5420	server shall change the value of PBS_O_QUEUE, even if the batch job is moved to another	
5421	queue.	
5422	A batch server that accepts a <i>Queue Batch Job Request</i> for a batch job for which the PBS_JOBID	
5423	value is missing from the value of the <i>Variable_List</i> attribute shall add that variable to the list and	
5424	set the value to the batch job identifier assigned by the server in the format:	
5425	<code>sequence_number.server</code>	
5426	A batch server that accepts a <i>Queue Batch Job Request</i> for a batch job for which the	
5427	PBS_JOBNAME value is missing from the value of the <i>Variable_List</i> attribute of the batch job	
5428	shall add that variable to the list and set the value to the <i>Job_Name</i> attribute of the batch job.	
5429	3.2.3.9 Batch Queue Status Request	
5430	A batch client can request that a batch server respond with the status and attributes of a queue.	
5431	Such a request is called a <i>Batch Queue Status Request</i> .	
5432	A batch server shall reject a <i>Batch Queue Status Request</i> if any of the following statements are true:	
5433	<ul style="list-style-type: none"> • The user of the batch client is not authorized to query the status of the designated queue. 	
5434	<ul style="list-style-type: none"> • The designated queue does not exist on the batch server. 	
5435	A batch server may reject a <i>Batch Queue Status Request</i> for other reasons. The conformance	
5436	document for an implementation shall describe the reasons for which a <i>Batch Queue Status</i>	
5437	<i>Request</i> is rejected. The conformance document for an implementation shall describe the method	
5438	used to determine whether the user of a client is authorized to perform the requested action.	
5439	A batch server that accepts a <i>Batch Queue Status Request</i> shall return a <i>Batch Queue Status Reply</i> to	
5440	the batch client.	

5441 3.2.3.10 *Release Batch Job Request*

5442 A batch client can request that server remove one or more holds from a batch job. Such a request
5443 is called a *Release Batch Job Request*.

5444 A batch server shall reject a *Release Batch Job Request* if any of the following statements are true:

- 5445 • The user of the batch client is not authorized to remove one or more of the requested holds
5446 from the batch job.
- 5447 • The batch server does not manage the specified job.

5448 A batch server may reject a *Release Batch Job Request* for other reasons. The conformance
5449 document for an implementation shall document the reasons for which a *Release Batch Job*
5450 *Request* may be rejected. The conformance document for an implementation shall describe the
5451 method used to determine whether the user of a client is authorized to perform the requested
5452 action.

5453 A batch server that accepts a *Release Batch Job Request* shall remove each type of hold listed in the
5454 *Release Batch Job Request*, that is present, from the value of the *Hold_Types* attribute of the batch
5455 job.

5456 3.2.3.11 *Rerun Batch Job Request*

5457 To rerun a batch job is to kill the session leader of the batch job and leave the batch job eligible
5458 for re-execution. A batch client can request that a batch server rerun a batch job. Such a request is
5459 called *Rerun Batch Job Request*.

5460 A batch server shall reject a *Rerun Batch Job Request* if any of the following statements are true:

- 5461 • The user of the batch client is not authorized to rerun the designated job.
- 5462 • The *Rerunable* attribute of the designated job has the value FALSE.
- 5463 • The designated job is not in the RUNNING state.
- 5464 • The batch server does not manage the designated job.

5465 A batch server may reject a *Rerun Batch Job Request* for other reasons. The conformance document
5466 for an implementation shall describe the reasons for which a *Rerun Batch Job Request* may be
5467 rejected. The conformance document for an implementation shall describe the method used to
5468 determine whether the user of a client is authorized to perform the requested action.

5469 A batch server that rejects a *Rerun Batch Job Request* shall in no way modify the execution of the
5470 batch job.

5471 A batch server that accepts a request to rerun a batch job shall perform the following services:

- 5472 • Requeue the batch job in the execution queue in which it was executing.
- 5473 • Send a SIGKILL signal to the process group of the session leader of the batch job.

5474 An implementation may indicate to the batch job owner that the batch job has been rerun. The
5475 conformance document for an implementation shall state whether the batch job owner is
5476 notified that a batch job is rerun, and if so, shall describe the means used.

5477 A batch server that reruns a batch job may send other signals to the session leader of the batch
5478 job prior to sending the SIGKILL signal. The conformance document for an implementation
5479 shall describe any other signals that may be sent.

5480 A batch server may preferentially select a rerun job for execution. The conformance document
5481 for an implementation shall state whether rerun jobs shall be selected for execution before other

5482 jobs.

5483 3.2.3.12 *Select Batch Jobs Request*

5484 A batch client can request from a batch server a list of jobs managed by that server that match a
5485 list of selection criteria. Such a request is called a *Select Batch Jobs Request*. All the batch jobs
5486 managed by the batch server that receives the request are candidates for selection.

5487 A batch server that accepts a *Select Batch Jobs Request* shall return a list of zero or more job
5488 identifiers that correspond to jobs that meet the selection criteria.

5489 If the batch client is not authorized to query the status of a batch job, the batch server shall not
5490 select the batch job.

5491 3.2.3.13 *Server Shutdown Request*

5492 A batch server is defined to have shut down when it does not respond to requests from clients
5493 and does not perform deferred services for jobs. A batch client can request that a batch server
5494 shut down. Such a request is called a *Server Shutdown Request*.

5495 A batch server shall reject a *Server Shutdown Request* from a client that is not authorized to shut
5496 down the batch server. The conformance document for an implementation shall describe the
5497 method used to determine whether the user of a client is authorized to perform the requested
5498 action.

5499 A batch server may reject a *Server Shutdown Request* for other reasons. The conformance
5500 document for an implementation shall document the reasons for which a *Server Shutdown*
5501 *Request* may be rejected.

5502 At server shutdown, a batch server shall do, in order of preference, one of the following:

- 5503 • If checkpointing is implemented and the batch job is checkpointable, then checkpoint the
5504 batch job and requeue it.
- 5505 • If the batch job is rerunnable, then requeue the batch job to be rerun (restarted from the
5506 beginning).
- 5507 • Abort the batch job.

5508 3.2.3.14 *Server Status Request*

5509 A batch client can request that a batch server respond with the status and attributes of the batch
5510 server. Such a request is called a *Server Status Request*.

5511 A batch server shall reject a *Server Status Request* if the following statement is true:

- 5512 • The user of the batch client is not authorized to query the status of the designated server.

5513 A batch server may reject a *Server Status Request* for other reasons. The conformance document
5514 for an implementation shall describe the reasons for which a *Server Status Request* may be
5515 rejected. The conformance document for an implementation shall describe the method used to
5516 determine whether the user of a client is authorized to perform the requested action.

5517 A batch server that accepts a *Server Status Request* shall return a *Server Status Reply* to the batch
5518 client.

5519 3.2.3.15 *Signal Batch Job Request*

5520 A batch client can request that a batch server signal the session leader of a batch job. Such a
5521 request is called a *Signal Batch Job Request*.

5522 A batch server shall reject a *Signal Batch Job Request* if any of the following statements are true:

- 5523 • The user of the batch client is not authorized to signal the batch job.
- 5524 • The job is not in the RUNNING state.
- 5525 • The batch server does not manage the designated job.
- 5526 • The requested signal is not supported by the implementation.

5527 A batch server may reject a *Signal Batch Job Request* for other reasons. The conformance
5528 document for an implementation shall describe the reasons for which a *Signal Batch Job Request*
5529 may be rejected. The conformance document for an implementation shall describe the method
5530 used to determine whether the user of a client is authorized to perform the requested action.

5531 A batch server that accepts a request to signal a batch job shall send the signal requested by the
5532 batch client to the process group of the session leader of the batch job.

5533 3.2.3.16 *Track Batch Job Request*

5534 *Track Batch Job Request* is an optional feature of batch servers. If an implementation supports
5535 *Track Batch Job Request*, the statements in this section apply and the configuration variable
5536 POSIX2_PBS_TRACK shall be set to 1.

5537 *Track Batch Job Request* provides a method for tracking the current location of a batch job. Clients
5538 may use the tracking information to determine the batch server that should receive a batch
5539 server request.

5540 If *Track Batch Job Request* is supported by a batch server, then when the batch server queues a
5541 batch job as a result of a *Queue Batch Job Request*, and the batch server is not the batch server that
5542 created the batch job, the batch server shall send a *Track Batch Job Request* to the batch server that
5543 created the job.

5544 If *Track Batch Job Request* is supported by a batch server, then the *Track Batch Job Request* may also
5545 be sent to other servers as a backup to the primary server. The method by which backup servers
5546 are specified is implementation-dependent.

5547 If *Track Batch Job Request* is supported by a batch server that receives a *Track Batch Job Request*,
5548 then the batch server shall record the current location of the batch job as contained in the
5549 request.

5550 3.3 Common Behavior for Batch Environment Utilities

5551 3.3.1 Batch Job Identifier

5552 A utility shall recognize *job_identifiers* of the format:

5553 `sequence_number[.server_name][@server]`

5554 where:

5555 *sequence_number* An integer that, when combined with *server_name*, provides a batch job
5556 identifier that is unique within the batch system.

5557 *server_name* The name of the batch server to which the batch job was originally submitted.

5558 *server* The name of the batch server that is currently managing the batch job.

5559 If the application omits the batch *server_name* portion of a batch job identifier, a utility shall use
5560 the name of a default batch server.

5561 If the application omits the batch *server* portion of a batch job identifier, a utility shall use:

- 5562 • The batch server indicated by *server_name*, if present.
- 5563 • The name of the default batch server.
- 5564 • The name of the batch server that is currently managing the batch job.

5565 If only *@server* is specified, then the status of all jobs owned by the user on the requested server
5566 is listed.

5567 The means by which a utility determines the default batch server is implementation-dependent.

5568 If the application presents the batch *server* portion of a batch job identifier to a utility, the utility
5569 shall send the request to the specified server.

5570 A strictly conforming application shall use the syntax described for the job identifier. Whenever
5571 a batch job identifier is specified whose syntax is not recognized by an implementation, then a
5572 message for each error that occurs shall be written to standard error and the utility shall exit
5573 with an exit status greater than zero.

5574 When a batch job identifier is supplied as an argument to a batch utility and the *server_name*
5575 portion of the batch job identifier is omitted, then the utility shall use the name of the default
5576 batch server.

5577 When a batch job identifier is supplied as an argument to a batch utility and the batch *server*
5578 portion of the batch job identifier is omitted, then the utility shall use either:

- 5579 • The name of the default batch server

5580 or:

- 5581 • The name of the batch server that is currently managing the batch job

5582 When a batch job identifier is supplied as an argument to a batch utility and the batch *server*
5583 portion of the batch job identifier is specified, then the utility shall send the required *Batch Server*
5584 *Request* to the specified server.

5585 3.3.2 Destination

5586 The utility shall recognize a *destination* of the format:

5587 [queue][@server]

5588 where:

5589 *queue* The name of a valid execution or routing queue at the batch server denoted by
5590 @*server*, defined as a string of up to 15 alphanumeric characters in the portable
5591 character set (see the System Interface Definitions volume of
5592 IEEE Std. 1003.1-200x, Section 6.1, Portable Character Set) where the first
5593 character is alphabetic.

5594 *server* The name of a batch server, defined as a string of alphanumeric characters in
5595 the portable character set.

5596 If the application omits the batch *server* portion of a destination, then the utility shall use either:

- 5597 • The name of the default batch server

5598 or:

- 5599 • The name of the batch server that is currently managing the batch job

5600 The means by which a utility determines the default batch server is implementation-dependent.

5601 If the application omits the *queue* portion of a destination, then the utility shall use the name of
5602 the default queue at the batch server chosen.

5603 The means by which a batch server determines its default queue is implementation-dependent.

5604 If a destination is specified in the *queue*@*server* form, then the utility shall use the specified queue
5605 at the specified server.

5606 A strictly conforming application shall use the syntax described for a destination. Whenever a
5607 destination is specified whose syntax is not recognized by an implementation, then a message
5608 shall be written to standard error and the utility shall exit with an exit status greater than zero.

5609 3.3.3 Multiple Keyword-Value Pairs

5610 For each option that can have multiple keyword-value pair arguments, the following rules shall
5611 apply. Examples of options that can have list-oriented option-arguments are **-u value@keyword**
5612 and **-l keyword=value**.

- 5613 1. If a batch utility is presented with a list-oriented option-argument for which a keyword has
5614 a corresponding value that begins with a single or double quote, then the utility shall stop
5615 interpreting the input stream for delimiters until a second single or double quote,
5616 respectively, is encountered. This feature allows some flexibility for a comma (',') or
5617 equals sign ('=') to be part of the value string for a particular keyword; for example:

5618 keywd1='val1,val2',keywd2="val3,val4"

5619 **Note:** This may require the user to escape the quotes as in the following command:

5620 foo -xkeywd1=\'val1,val2\',keywd2=\"val3,val4\"

- 5621 2. If a batch server is presented with a list-oriented attribute that has a keyword that was
5622 encountered earlier in the list, then the later entry for that keyword shall replace the earlier
5623 entry.
- 5624 3. If a batch server is presented with a list-oriented attribute that has a keyword without any
5625 corresponding value of the form *keyword=* or *@keyword* and the same keyword was

- 5626 encountered earlier in the list, then the prior entry for that keyword shall be ignored by the
5627 batch server.
- 5628 4. If a batch utility is expecting a list-oriented option-argument entry of the form
5629 *keyword=value*, but is presented with an entry of the form *keyword* without any
5630 corresponding *value*, then the entry shall be treated as though a default value of NULL was
5631 assigned (that is, *keyword=NULL*) for entry parsing purposes. The utility shall include only
5632 the keyword, not the NULL value, in the associated job attribute.
- 5633 5. If a batch utility is expecting a list-oriented option-argument entry of the form
5634 *value@keyword*, but is presented with an entry of the form *value* without any corresponding
5635 *keyword*, then the entry shall be treated as though a keyword of NULL was assigned (that
5636 is, *value@NULL*) for entry parsing purposes. The utility shall include only the value, not
5637 the NULL keyword, in the associated job attribute.
- 5638 6. A batch server shall accept a list-oriented attribute that has multiple occurrences of the
5639 same keyword, interpreting the keywords, in order, with the last value encountered taking
5640 precedent over prior instances of the same keyword. This rule allows, but does not require,
5641 a batch utility to preprocess the attribute to remove duplicate keywords.
- 5642 7. If a batch utility is presented with multiple list-oriented option-arguments on the
5643 command line or in script directives, or both, for a single option, then the utility shall
5644 concatenate, in order, any command line keyword and value pairs to the end of any
5645 directive keyword and value pairs separated by a single comma to produce a single string
5646 that is an equivalent, valid option-argument. The resulting string shall be assigned to the
5647 associated attribute of the batch job (after optionally removing duplicate entries as
5648 described in item 6.
5649

Utilities

5650

5651

This chapter contains the definitions of the utilities, as follows:

5652

- Mandatory utilities that are present on every conformant system

5653

5654

5655

- Optional utilities that are present only on systems supporting the associated option; see Section 1.8.1 on page 14 for information on the options in this volume of IEEE Std. 1003.1-200x

5656 NAME

5657 admin — create and administer SCCS files (DEVELOPMENT)

5658 SYNOPSIS

```

5659 xSI admin -i[name][-n][-a login][-d flag][-f flag][-m mrlist][-r rel]
5660      [-t[name][-y[comment]] newfile

5661      admin -n[-a login][-d flag][-f flag][-m mrlist][-t[name]][-y[comment]]
5662      newfile ...

5663      admin [-a login][-d flag][-m mrlist][-r rel][-t[name]] file ...

5664      admin -h file ...

5665      admin -z file ...
5666

```

5667 DESCRIPTION

5668 The *admin* utility shall create new SCCS files or change parameters of existing ones. If a named
 5669 file does not exist, it shall be created, and its parameters shall be initialized according to the
 5670 specified options. Parameters not initialized by an option shall be assigned a default value. If a
 5671 named file does exist, parameters corresponding to specified options shall be changed, and other
 5672 parameters shall be left as is.

5673 All SCCS file names supplied by the application shall be of the form *s.filename*. New SCCS files
 5674 shall be given read-only permission mode. Write permission in the parent directory is required
 5675 to create a file. All writing done by *admin* shall be to a temporary *x-file*, named *x.filename* (see *get*)
 5676 created with read-only mode if *admin* is creating a new SCCS file, or created with the same mode
 5677 as that of the SCCS file if the file already exists. After successful execution of *admin*, the SCCS file
 5678 shall be removed (if it exists), and the *x-file* shall be renamed with the name of the SCCS file. This
 5679 ensures that changes are made to the SCCS file only if no errors occur.

5680 The *admin* utility shall also use a transient lock file (named *z.filename*), which is used to prevent
 5681 simultaneous updates to the SCCS file; see *get* on page 510.

5682 OPTIONS

5683 The *admin* utility shall conform to the System Interface Definitions volume of
 5684 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines, except that the *-i*, *-t*, and *-y*
 5685 options have optional option-arguments. These optional option-arguments cannot be presented
 5686 as separate arguments. The following options are supported:

5687 **-n** Create a new SCCS file. When *-n* is used without *-i*, the SCCS file is created with
 5688 control information but without any file data.

5689 **-i[name]** Specify the *name* of a file from which the text for a new SCCS file is to be taken. The
 5690 text constitutes the first delta of the file (see the *-r* option for delta numbering
 5691 scheme). If the *-i* option is used, but the *name* option-argument is omitted, the text
 5692 is obtained by reading the standard input. If this option is omitted, the SCCS file is
 5693 created with control information but without any file data. The *-i* option implies
 5694 the *-n* option.

5695 **-r rel** Specify the *release* into which the initial delta is inserted. If the *-r* option is not
 5696 used, the initial delta is inserted into release 1. The level of the initial delta is
 5697 always 1 (by default, initial deltas are named 1.1).

5698 **-t[name]** Specify the *name* of a file from which descriptive text for the SCCS file is to be
 5699 taken. In the case of existing SCCS files (neither *-i* nor *-n* is specified):

5700		<ul style="list-style-type: none"> • A -t option without a <i>name</i> option-argument causes the removal of descriptive text (if any) currently in the SCCS file.
5701		
5702		<ul style="list-style-type: none"> • A -t option with a <i>name</i> option-argument causes the text (if any) in the named file to replace the descriptive text (if any) currently in the SCCS file.
5703		
5704	-f flag	Specify a <i>flag</i> , and, possibly, a value for the <i>flag</i> , to be placed in the SCCS file.
5705		Several -f options may be supplied on a single <i>admin</i> command line. The allowable flags and their values are:
5706		
5707	b	Allow use of the -b option on a <i>get</i> command to create branch deltas.
5708	cceil	Specify the highest release (that is, ceiling), a number less than or equal to 9 999, which may be retrieved by a <i>get</i> command for editing. The default value for an unspecified c flag is 9 999.
5709		
5710		
5711	ffloor	Specify the lowest release (that is, floor), a number greater than 0 but less than 9 999, which may be retrieved by a <i>get</i> command for editing. The default value for an unspecified f flag is 1.
5712		
5713		
5714	dSID	Specify the default delta number (SID) to be used by a <i>get</i> command.
5715	istr	Treat the “No ID keywords” message issued by <i>get</i> or <i>delta</i> as a fatal error. In the absence of this flag, the message is only a warning. The message is issued if no SCCS identification keywords (see <i>get</i> on page 510) are found in the text retrieved or stored in the SCCS file. If a value is supplied, the application shall ensure that the keywords exactly match the given string; however, the string shall contain a keyword, and no embedded <newline>s.
5716		
5717		
5718		
5719		
5720		
5721		
5722	j	Allow concurrent <i>get</i> commands for editing on the same SID of an SCCS file. This allows multiple concurrent updates to the same version of the SCCS file.
5723		
5724		
5725	l list	Specify a <i>list</i> of releases to which deltas can no longer be made (that is, <i>get</i> -e against one of these locked releases fails). The <i>list</i> has the following syntax:
5726		
5727		
5728		<list> ::= <range> <list>, <range>
5729		<range> ::= SID a
5730		The character <i>a</i> in the <i>list</i> is equivalent to specifying all releases for the named SCCS file.
5731		
5732	n	Cause <i>delta</i> to create a null delta in each of those releases (if any) being skipped when a delta is made in a new release (for example, in making delta 5.1 after delta 2.7, releases 3 and 4 are skipped). These null deltas serve as anchor points so that branch deltas may later be created from them. The absence of this flag causes skipped releases to be nonexistent in the SCCS file, preventing branch deltas from being created from them in the future.
5733		
5734		
5735		
5736		
5737		
5738		
5739	qtext	Substitute user-definable <i>text</i> for all occurrences of the %Q% keyword in the SCCS file text retrieved by <i>get</i> .
5740		
5741	mmod	Specify the module name of the SCCS file substituted for all occurrences of the %M% keyword in the SCCS file text retrieved by <i>get</i> . If the m flag is not specified, the value assigned is the name of the SCCS file with the leading ‘.’ removed.
5742		
5743		
5744		

5745 **ttype** Specify the *type* of module in the SCCS file substituted for all occurrences
5746 of the %Y% keyword in the SCCS file text retrieved by *get*.

5747 **vpgm** Cause *delta* to prompt for modification request (MR) numbers as the
5748 reason for creating a delta. The optional value specifies the name of an
5749 MR number validation program. (If this flag is set when creating an SCCS
5750 file, the application shall ensure that the **m** option is also used even if its
5751 value is null.)

5752 **-d flag** Remove (delete) the specified *flag* from an SCCS file. Several **-d** options may be
5753 supplied on a single *admin* command. See the **-f** option for allowable *flag* names.
5754 (The **l**list flag gives a *list* of releases to be unlocked. See the **-f** option for further
5755 description of the **l** flag and the syntax of a *list*.)

5756 **-a login** Specify a *login* name, or numerical group ID, to be added to the list of users who
5757 may make deltas (changes) to the SCCS file. A group ID is equivalent to specifying
5758 all *login* names common to that group ID. Several **-a** options may be used on a
5759 single *admin* command line. As many *logins*, or numerical group IDs, as desired
5760 may be on the list simultaneously. If the list of users is empty, then anyone may
5761 add deltas. If *login* or group ID is preceded by a '!', the users so specified are
5762 denied permission to make deltas.

5763 **-e login** Specify a *login* name, or numerical group ID, to be erased from the list of users
5764 allowed to make deltas (changes) to the SCCS file. Specifying a group ID is
5765 equivalent to specifying all *login* names common to that group ID. Several **-e**
5766 options may be used on a single *admin* command line.

5767 **-y[comment]** Insert the *comment* text into the SCCS file as a comment for the initial delta in a
5768 manner identical to that of *delta*. In the POSIX locale, omission of the **-y** option
5769 results in a default comment line being inserted in the form:

5770 "date and time created %s %s by %s", <date>, <time>, <login>
5771 where <date> is expressed in the *date* utility's %y/%m/%d format, <time> in the
5772 *date* utility's %T format, and <login> is the login name of the user creating the file.

5773 **-m mrlist** Insert the list of modification request (MR) numbers into the SCCS file as the
5774 reason for creating the initial delta in a manner identical to *delta*. The application
5775 shall ensure that the **v** flag is set and the MR numbers are validated if the **v** flag has
5776 a value (the name of an MR number validation program). Diagnostics occur if the
5777 **v** flag is not set or MR validation fails.

5778 **-h** Check the structure of the SCCS file and compare the newly computed checksum
5779 (the sum of all the characters in the SCCS file except those in the first line) with the
5780 checksum that is stored in the first line of the SCCS file. Appropriate error
5781 diagnostics are produced.

5782 **-z** Recompute the SCCS file checksum and store it in the first line of the SCCS file (see
5783 the **-h** option above). Note that use of this option on a truly corrupted file may
5784 prevent future detection of the corruption.

5785 OPERANDS

5786 The following operands shall be supported:

5787 **file** A path name of an existing SCCS file or a directory. If *file* is a directory, *admin*
5788 behaves as though each file in the directory were specified as a named file, except
5789 that non-SCCS files (last component of the path name does not begin with **s**.) and
5790 unreadable files are silently ignored.

5791 *newfile* A path name of an SCCS file to be created.

5792 If a single instance of *file* or *newfile* is specified as '*-*', the standard input is read; each line of the
5793 standard input is taken to be the name of an SCCS file to be processed. Non-SCCS files and
5794 unreadable files are silently ignored.

5795 **STDIN**

5796 The standard input shall be a text file used only if the *-i* is specified without an option-argument
5797 or if a *file* or *newfile* operand is specified as '*-*'. If the first character of any standard input line is
5798 SOH (binary 001), the results are unspecified.

5799 **INPUT FILES**

5800 The existing SCCS files are text files of an unspecified format. The file named by the *-i* option's
5801 *name* option-argument is a text file; if the first character of any line in this file is SOH (binary
5802 001), the results are unspecified.

5803 **ENVIRONMENT VARIABLES**

5804 The following environment variables shall affect the execution of *admin*:

5805 *LANG* Provide a default value for the internationalization variables that are unset or null.
5806 If *LANG* is unset or null, the corresponding value from the implementation-
5807 dependent default locale shall be used. If any of the internationalization variables
5808 contains an invalid setting, the utility shall behave as if none of the variables had
5809 been defined.

5810 *LC_ALL* If set to a non-empty string value, override the values of all the other
5811 internationalization variables.

5812 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
5813 characters (for example, single-byte as opposed to multi-byte characters in
5814 arguments and input files).

5815 *LC_MESSAGES*

5816 Determine the locale that should be used to affect the format and contents of
5817 diagnostic messages written to standard error and the contents of the default *-y*
5818 comment.

5819 *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.

5820 **ASYNCHRONOUS EVENTS**

5821 Default.

5822 **STDOUT**

5823 Not used.

5824 **STDERR**

5825 Used only for diagnostic messages.

5826 **OUTPUT FILES**

5827 Any SCCS files created shall be text files of an unspecified format. During processing of a *file*, a
5828 locking *z-file*, as described in *get* on page 510, may be created and deleted.

5829 **EXTENDED DESCRIPTION**

5830 None.

5831 **EXIT STATUS**

5832 The following exit values shall be returned:

5833 0 Successful completion.

5834 >0 An error occurred.

5835 **CONSEQUENCES OF ERRORS**

5836 Default.

5837 **APPLICATION USAGE**

5838 It is recommended that directories containing SCCS files be writable by the owner only, and that
5839 SCCS files themselves be read-only. The mode of the directories should allow only the owner to
5840 modify SCCS files contained in the directories. The mode of the SCCS files prevents any
5841 modification at all except by SCCS commands.

5842 **EXAMPLES**

5843 None.

5844 **RATIONALE**

5845 None.

5846 **FUTURE DIRECTIONS**

5847 A version of *admin* that fully supports the System Interface Definitions volume of
5848 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines may be introduced in a future
5849 issue.

5850 **SEE ALSO**

5851 *delta, get, prs, what*

5852 **CHANGE HISTORY**

5853 First released in Issue 2.

5854 **Issue 4**

5855 Format reorganized.

5856 Conformance to Utility Syntax Guidelines mandated, with exceptions as noted.

5857 Internationalized environment variable support mandated.

5858 **Issue 6**

5859 The normative text is reworded to avoid use of the term “must” for application requirements.

5860 **NAME**

5861 alias — define or display aliases

5862 **SYNOPSIS**5863 UP alias [*alias-name*[=*string*] ...]
58645865 **DESCRIPTION**5866 The *alias* utility shall create or redefine alias definitions or write the values of existing alias
5867 definitions to standard output. An alias definition provides a string value that shall replace a
5868 command name when it is encountered; see Section 2.3.1 on page 40.5869 An alias definition shall affect the current shell execution environment and the execution
5870 environments of the subshells of the current shell. When used as specified by this volume of
5871 IEEE Std. 1003.1-200x, the alias definition shall not affect the parent process of the current shell
5872 nor any utility environment invoked by the shell; see Section 2.12 on page 90.5873 **OPTIONS**

5874 None.

5875 **OPERANDS**

5876 The following operands shall be supported:

5877 *alias-name* Write the alias definition to standard output.5878 *alias-name=string*5879 Assign the value of *string* to the alias *alias-name*.

5880 If no operands are given, all alias definitions shall be written to standard output.

5881 **STDIN**

5882 Not used.

5883 **INPUT FILES**

5884 None.

5885 **ENVIRONMENT VARIABLES**5886 The following environment variables shall affect the execution of *alias*:5887 *LANG* Provide a default value for the internationalization variables that are unset or null.
5888 If *LANG* is unset or null, the corresponding value from the implementation-
5889 dependent default locale shall be used. If any of the internationalization variables
5890 contains an invalid setting, the utility shall behave as if none of the variables had
5891 been defined.5892 *LC_ALL* If set to a non-empty string value, override the values of all the other
5893 internationalization variables.5894 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
5895 characters (for example, single-byte as opposed to multi-byte characters in
5896 arguments).5897 *LC_MESSAGES*5898 Determine the locale that should be used to affect the format and contents of
5899 diagnostic messages written to standard error.5900 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.

5901 **ASYNCHRONOUS EVENTS**

5902 Default.

5903 **STDOUT**5904 The format for displaying aliases (when no operands or only *name* operands are specified) shall
5905 be:5906 "%s=%s\n", *name*, *value*5907 The *value* string shall be written with appropriate quoting so that it is suitable for reinput to the
5908 shell. See the description of shell quoting in Section 2.2 on page 36.5909 **STDERR**

5910 Used only for diagnostic messages.

5911 **OUTPUT FILES**

5912 None.

5913 **EXTENDED DESCRIPTION**

5914 None.

5915 **EXIT STATUS**

5916 The following exit values shall be returned:

5917 0 Successful completion.

5918 >0 One of the *name* operands specified did not have an alias definition, or an error occurred.5919 **CONSEQUENCES OF ERRORS**

5920 Default.

5921 **APPLICATION USAGE**5922 Application writers should note that this utility need not be provided on systems that do not
5923 support the User Portability Utilities option.5924 **EXAMPLES**5925 1. Change *ls* to give a columnated, more annotated output:5926 `alias ls="ls -CF"`

5927 2. Create a simple “redo” command to repeat previous entries in the command history file:

5928 `alias r='fc -s'`5929 3. Use 1K units for *du*:5930 `alias du=du\ -k`5931 4. Set up *nohup* so that it can deal with an argument that is itself an alias name:5932 `alias nohup="nohup "`5933 **RATIONALE**5934 The *alias* description is based on historical KornShell implementations. Known differences exist
5935 between that and the C shell. The KornShell version was adopted to be consistent with all the
5936 other KornShell features in this volume of IEEE Std. 1003.1-200x, such as command line editing.5937 Since *alias* affects the current shell execution environment, it is generally provided as a shell
5938 regular built-in.5939 Historical versions of the KornShell have allowed aliases to be exported to scripts that are
5940 invoked by the same shell. This is triggered by the *alias -x* flag; it is allowed by this volume of
5941 IEEE Std. 1003.1-200x only when an explicit extension such as *-x* is used. The standard

5942 developers considered that aliases were of use primarily to interactive users and that they
5943 should normally not affect shell scripts called by those users; functions are available to such
5944 scripts.

5945 Historical versions of the KornShell had not written aliases in a quoted manner suitable for
5946 reentry to the shell, but this volume of IEEE Std. 1003.1-200x has made this a requirement for all
5947 similar output. Therefore, consistency with this volume of IEEE Std. 1003.1-200x was chosen
5948 over this detail of historical practice.

5949 **FUTURE DIRECTIONS**

5950 None.

5951 **SEE ALSO**

5952 Section 2.9.5 on page 79

5953 **CHANGE HISTORY**

5954 First released in Issue 4.

5955 **Issue 6**

5956 This utility is now marked as part of the User Portability Utilities option.

5957 The APPLICATION USAGE section is added.

5958 NAME

5959 ar — create and maintain library archives

5960 SYNOPSIS

5961 SD ar -d[-v] archive file ...

5962

5963 XSI ar -m[-abiv][posname] archive file ...

5964

5965 XSI ar -p[-v][-s]archive [file ...]

5966 XSI ar -q[-cv] archive file ...

5967

5968 XSI ar -r[-cuv][-abi][posname]archive file ...

5969 XSI ar -t[-v][-s]archive [file ...]

5970 XSI ar -x[-v][-sCT]archive [file ...]

5971 DESCRIPTION

5972 The *ar* utility can be used to create and maintain groups of files combined into an archive. Once
 5973 an archive has been created, new files can be added, and existing files can be extracted, deleted,
 5974 or replaced. When an archive consists entirely of valid object files, the implementation shall
 5975 format the archive so that it is usable as a library for link editing (see *c89*, *cc*, and *fort77*). When
 5976 some of the archived files are not valid object files, the suitability of the archive for library use is
 5977 XSI undefined. If an archive file consists entirely of printable files, the entire archive file is printable.

5978 When *ar* creates an archive file, it creates administrative information indicating whether a
 5979 symbol table is present in the archive. When there is at least one object file that *ar* recognizes as
 5980 such in the archive, an archive symbol table is created in the archive file and maintained by *ar*; it
 5981 is used by the link editor to search the archive file. Whenever the *ar* utility is used to create or
 5982 update the contents of such an archive, the symbol table is rebuilt. The *-s* option forces the
 5983 symbol table to be rebuilt.

5984 All *file* operands can be path names. However, files within archives shall be named by a file
 5985 name, which is the last component of the path name used when the file was entered into the
 5986 archive. The comparison of *file* operands to the names of files in archives shall be performed by
 5987 comparing the last component of the operand to the name of the archive file.

5988 It is unspecified whether multiple files in the archive may be identically named. In the case of
 5989 XSI such files, however, each *file* and *posname* operand shall match only the first archive file having a
 5990 name that is the same as the last component of the operand.

5991 OPTIONS

5992 The *ar* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,
 5993 Section 12.2, Utility Syntax Guidelines.

5994 The following options shall be supported:

5995 XSI **-a** Position new files in the archive after the file named by the *posname* operand.

5996 XSI **-b** Position new files in the archive before the file named by the *posname* operand.

5997 **-c** Suppress the diagnostic message that is written to standard error by default when
 5998 the archive file *archive* is created.

5999 XSI **-C** Prevent extracted files from replacing like-named files in the file system. This
 6000 option is useful when *-T* is also used, to prevent truncated file names from
 6001 replacing files with the same prefix.

6002		-d	Delete one or more <i>files</i> from <i>archive</i> .
6003	XSI	-i	Position new files in the archive before the file named by the <i>posname</i> operand (equivalent to -b).
6004			
6005	XSI	-m	Move the named files. The -a , -b , or -i options with the <i>posname</i> operand indicate the position; otherwise, move the files to the end of the archive.
6006			
6007		-p	Write the contents of the <i>files</i> from <i>archive</i> to the standard output. If no <i>files</i> are specified, the contents of all files in the archive shall be written in the order of the archive.
6008			
6009			
6010	XSI	-q	Quickly append the named files to the end of the archive file. In this case <i>ar</i> does not check whether the added members are already in the archive. This is useful to bypass the searching otherwise done when creating a large archive piece by piece.
6011			
6012			
6013		-r	Replace or add <i>files</i> to <i>archive</i> . If the archive named by <i>archive</i> does not exist, a new archive file shall be created and a diagnostic message shall be written to standard error (unless the -c option is specified). If no <i>files</i> are specified and the <i>archive</i> exists, the results are undefined. Files that replace existing files shall not change the order of the archive. Files that do not replace existing files shall be appended to the archive unless a -a , -b , or -i option specifies another position.
6014			
6015			
6016			
6017			
6018	XSI		
6019	XSI	-s	Force the regeneration of the archive symbol table even if <i>ar</i> is not invoked with an option that modifies the archive file contents. This option is useful to restore the archive symbol table after it has been stripped; see <i>strip</i> .
6020			
6021			
6022		-t	Write a table of contents of <i>archive</i> to the standard output. The files specified by the <i>file</i> operands shall be included in the written list. If no <i>file</i> operands are specified, all files in <i>archive</i> shall be included in the order of the archive.
6023			
6024			
6025	XSI	-T	Allow file name truncation of extracted files whose archive names are longer than the file system can support. By default, extracting a file with a name that is too long is an error; a diagnostic message is written and the file is not extracted.
6026			
6027			
6028		-u	Update older files. When used with the -r option, files within the archive are replaced only if the corresponding <i>file</i> has a modification time that is at least as new as the modification time of the file within the archive.
6029			
6030			
6031		-v	Give verbose output. When used with the option characters -d , -r , or -x , write a detailed file-by-file description of the archive creation and maintenance activity, as described in the STDOUT section.
6032			
6033			
6034			When used with -p , write the name of the file to the standard output before writing the file itself to the standard output, as described in the STDOUT section.
6035			
6036			When used with -t , include a long listing of information about the files within the archive, as described in the STDOUT section.
6037			
6038		-x	Extract the files named by the <i>file</i> operands from <i>archive</i> . The contents of the archive file shall not be changed. If no <i>file</i> operands are given, all files in the archive shall be extracted. The modification time of each file extracted shall be set to the time the file is extracted from the archive.
6039			
6040			
6041			

6042 OPERANDS

6043 The following operands shall be supported:

6044 *archive* A path name of the archive file.

6045 *file* A path name. Only the last component shall be used when comparing against the
6046 names of files in the archive. If two or more *file* operands have the same last path
6047 name component (basename), the results are unspecified. The implementation's
6048 archive format shall not truncate valid file names of files added to or replaced in
6049 the archive.

6050 XSI *posname* The name of a file in the archive file, used for relative positioning; see options **-m**
6051 and **-r**.

6052 **STDIN**
6053 Not used.

6054 **INPUT FILES**
6055 The input file named by *archive* shall be a file in the format created by *ar -r*.

6056 **ENVIRONMENT VARIABLES**
6057 The following environment variables shall affect the execution of *ar*:

6058 *LANG* Provide a default value for the internationalization variables that are unset or null.
6059 If *LANG* is unset or null, the corresponding value from the implementation-
6060 dependent default locale shall be used. If any of the internationalization variables
6061 contains an invalid setting, the utility shall behave as if none of the variables had
6062 been defined.

6063 *LC_ALL* If set to a non-empty string value, override the values of all the other
6064 internationalization variables.

6065 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
6066 characters (for example, single-byte as opposed to multi-byte characters in
6067 arguments and input files).

6068 *LC_MESSAGES*
6069 Determine the locale that should be used to affect the format and contents of
6070 diagnostic messages written to standard error.

6071 *LC_TIME* Determine the format and content for date and time strings written by *ar -tv*.

6072 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.

6073 *TMPDIR* Determine the path name that overrides the default directory for temporary files, if
6074 any.

6075 **ASYNCHRONOUS EVENTS**
6076 Default.

6077 **STDOUT**
6078 If the **-d** option is used with the **-v** option, the standard output format shall be:
6079 " d - %s\n", <file>
6080 where *file* is the operand specified on the command line.

6081 If the **-p** option is used with the **-v** option, *ar* shall precede the contents of each file with:
6082 "\n<%s>\n\n", <file>
6083 where *file* is the operand specified on the command line, if *file* operands were specified, and the
6084 name of the file in the archive if they were not.

6085 If the **-r** option is used with the **-v** option:

6086 • If *file* is already in the archive, the standard output format shall be: |

6087 "r - %s\n", <file> |

6088 where <file> is the operand specified on the command line. |

6089 • If *file* is not already in the archive, the standard output format shall be: |

6090 "a - %s\n", <file> |

6091 where <file> is the operand specified on the command line. |

6092 If the **-t** option is used, *ar* shall write the names of the files to the standard output in the format: |

6093 "%s\n", <file> |

6094 where *file* is the operand specified on the command line, if *file* operands were specified, or the

6095 name of the file in the archive if they were not.

6096 If the **-t** option is used with the **-v** option, the standard output format shall be:

6097 "%s %u/%u %u %s %d %d:%d %d %s\n", <member mode>, <user ID> ,

6098 <group ID>, <number of bytes in member> ,

6099 <abbreviated month>, <day-of-month>, <hour> ,

6100 <minute>, <year>, <file>

6101 where:

6102 <file> Shall be the operand specified on the command line, if *file* operands were specified,

6103 or the name of the file in the archive if they were not.

6104 <member

6105 *Shall be formatted the same as the <file mode> string defined in the STDOUT section of*

6106 *ls*, except that the first character, the <entry type>, is not used; the string represents

6107 the file mode of the archive member at the time it was added to or replaced in the

6108 archive.

6109 The following represent the last-modification time of a file when it was most recently added to

6110 or replaced in the archive:

6111 <abbreviated month>

6112 Equivalent to the %b format in *date*.

6113 <day-of-month>

6114 Equivalent to the %e format in *date*.

6115 <hour> Equivalent to the %H format in *date*.

6116 <minute> Equivalent to the %M format in *date*.

6117 <year> Equivalent to the %Y format in *date*.

6118 When *LC_TIME* does not specify the POSIX locale, a different format and order of presentation

6119 of these fields relative to each other may be used in a format appropriate in the specified locale.

6120 If the **-x** option is used with the **-v** option, the standard output format shall be:

6121 "x - %s\n", <file>

6122 where *file* is the operand specified on the command line, if *file* operands were specified, or the

6123 name of the file in the archive if they were not.

6124 STDERR

6125 Used only for diagnostic messages. The diagnostic message about creating a new archive when
6126 **-c** is not specified shall not modify the exit status.

6127 OUTPUT FILES

6128 Archives are files with unspecified formats.

6129 EXTENDED DESCRIPTION

6130 None.

6131 EXIT STATUS

6132 The following exit values shall be returned:

6133 0 Successful completion.

6134 >0 An error occurred.

6135 CONSEQUENCES OF ERRORS

6136 Default.

6137 APPLICATION USAGE

6138 None.

6139 EXAMPLES

6140 None.

6141 RATIONALE

6142 The archive format is not described. It is recognized that there are several known *ar* formats,
6143 which are not compatible. The *ar* utility is included, however, to allow creation of archives that
6144 are intended for use only on one machine. The archive file is specified as a file, and it can be
6145 moved as a file. This does allow an archive to be moved from one machine to another machine
6146 that uses the same implementation of *ar*.

6147 Utilities such as *pax* (and its forebears *tar* and *cpio*) also provide portable “archives”. This is a not
6148 a duplication; the *ar* utility is included to provide an interface primarily for *make* and the
6149 compilers, based on a historical model.

6150 In historical implementations, the **-q** option (available on XSI-conforming systems) is known to
6151 execute quickly because *ar* does not check on whether the added members are already in the
6152 archive. This is useful to bypass the searching otherwise done when creating a large archive
6153 piece-by-piece. These remarks may but need not remain true for a brand new implementation of
6154 this utility; hence, these remarks have been moved into the RATIONALE.

6155 BSD implementations historically required applications to provide the **-s** option whenever the
6156 archive was supposed to contain a symbol table. As in this volume of IEEE Std. 1003.1-200x,
6157 System V historically creates or updates an archive symbol table whenever an object file is
6158 removed from, added to, or updated in the archive.

6159 The OPERANDS section requires what might seem to be true without specifying it: the archive
6160 cannot truncate the file names below {NAME_MAX}. Some historical implementations do so,
6161 however, causing unexpected results for the application. Therefore, this volume of
6162 IEEE Std. 1003.1-200x makes the requirement explicit to avoid misunderstandings.

6163 According to the System V documentation, the options **-dmpqrtx** are not required to begin with
6164 a hyphen ('-'). This volume of IEEE Std. 1003.1-200x requires that a conforming application
6165 use the leading hyphen.

6166 The archive format used by the 4.4 BSD implementation is documented in this RATIONALE as
6167 an example:

6168 A file created by *ar* begins with the “magic” string “!*<arch>*\\n”. The rest of the archive is
6169 made up of objects, each of which is composed of a header for a file, a possible file name, and
6170 the file contents. The header is portable between machine architectures, and, if the file
6171 contents are printable, the archive is itself printable.

6172 The header is made up of six ASCII fields, followed by a two-character trailer. The fields are
6173 the object name (16 characters), the file last modification time (12 characters), the user and
6174 group IDs (each 6 characters), the file mode (8 characters), and the file size (10 characters). All
6175 numeric fields are in decimal, except for the file mode, which is in octal.

6176 The modification time is the file *st_mtime* field. The user and group IDs are the file *st_uid* and
6177 *st_gid* fields. The file mode is the file *st_mode* field. The file size is the file *st_size* field. The
6178 two-byte trailer is the string “*<newline>*”.

6179 Only the name field has any provision for overflow. If any file name is more than 16
6180 characters in length or contains an embedded space, the string “#1/” followed by the ASCII
6181 length of the name is written in the name field. The file size (stored in the archive header) is
6182 incremented by the length of the name. The name is then written immediately following the
6183 archive header.

6184 Any unused characters in any of these fields are written as *<space>* characters. If any fields
6185 are their particular maximum number of characters in length, there is no separation between
6186 the fields.

6187 Objects in the archive are always an even number of bytes long; files that are an odd number
6188 of bytes long are padded with a *<newline>* character, although the size in the header does
6189 not reflect this.

6190 The *ar* utility description requires that (when all its members are valid object files) *ar* produce an
6191 object code library, which the linkage editor can use to extract object modules. If the linkage
6192 editor needs a symbol table to permit random access to the archive, *ar* must provide it; however,
6193 *ar* does not require a symbol table.

6194 The BSD *-o* option was omitted. It is a rare portable application that uses *ar* to extract object
6195 code from a library with concern for its modification time, since this can only be of importance
6196 to *make*. Hence, since this functionality is not deemed important for applications portability, the
6197 modification time of the extracted files is set to the current time.

6198 There is at least one known implementation (for a small computer) that can accommodate only
6199 object files for that system, disallowing mixed object and other files. The ability to handle any
6200 type of file is not only historical practice for most implementations, but is also a reasonable
6201 expectation.

6202 Consideration was given to changing the output format of *ar -tv* to the same format as the
6203 output of *ls -l*. This would have made parsing the output of *ar* the same as that of *ls*. This was
6204 rejected in part because the current *ar* format is commonly used and changes would break
6205 historical usage. Second, *ar* gives the user ID and group ID in numeric format separated by a
6206 slash. Changing this to be the user name and group name would not be correct if the archive
6207 were moved to a machine that contained a different user database. Since *ar* cannot know
6208 whether the archive file was generated on the same machine, it cannot tell what to report.

6209 The text on the *-ur* option combination is historical practice—since one file name can easily
6210 represent two different files (for example, */a/foo* and */b/foo*), it is reasonable to replace the
6211 member in the archive even when the modification time in the archive is identical to that in the
6212 file system.

6213 **FUTURE DIRECTIONS**

6214 None.

6215 **SEE ALSO**

6216 *c89*, *pax*, *strip* the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 13, |
 6217 Headers, <**unistd.h**> description of {POSIX_NO_TRUNC} |

6218 **CHANGE HISTORY**

6219 First released in Issue 2.

6220 **Issue 4**

6221 Aligned with the ISO/IEC 9945-2:1993 standard.

6222 The **-C** and **-T** options are added.6223 **Issue 5**

6224 FUTURE DIRECTIONS section added.

6225 **Issue 6**

6226 This utility is now marked as part of the Software Development Utilities option. |

6227 The **STDOUT** description is changed for the **-v** option to align with the IEEE P1003.2b draft |
 6228 standard. |

6229 The normative text is reworded to avoid use of the term “must” for application requirements. |

6230 **NAME**

6231 asa — interpret carriage-control characters

6232 **SYNOPSIS**6233 FR asa [*file* ...]

6234

6235 **DESCRIPTION**6236 The *asa* utility shall write its input files to standard output, mapping carriage-control characters
6237 from the text files to line-printer control sequences in an implementation-dependent manner.6238 The first character of every line shall be removed from the input, and the following actions are
6239 performed:

6240 If the character removed is:

6241 <space> The rest of the line is output without change.

6242 0 A <newline> character is output, then the rest of the input line.

6243 1 One or more implementation-dependent characters that causes an advance to the next
6244 page shall be output, followed by the rest of the input line.6245 + The <newline> character of the previous line shall be replaced with one or more
6246 implementation-dependent characters that causes printing to return to column position
6247 1, followed by the rest of the input line. If the '+' is the first character in the input, it
6248 shall have the same effect as the <space> character.6249 The action of the *asa* utility is unspecified upon encountering any character other than those
6250 listed above as the first character in a line.6251 **OPTIONS**

6252 None.

6253 **OPERANDS**6254 *file* A path name of a text file used for input. If no *file* operands are specified, the
6255 standard input shall be used.6256 **STDIN**6257 The standard input is used only if no *file* operands are specified; see the INPUT FILES section.6258 **INPUT FILES**

6259 The input files shall be text files.

6260 **ENVIRONMENT VARIABLES**6261 The following environment variables shall affect the execution of *asa*:6262 *LANG* Provide a default value for the internationalization variables that are unset or null.
6263 If *LANG* is unset or null, the corresponding value from the implementation-
6264 dependent default locale shall be used. If any of the internationalization variables
6265 contains an invalid setting, the utility shall behave as if none of the variables had
6266 been defined.6267 *LC_ALL* If set to a non-empty string value, override the values of all the other
6268 internationalization variables.6269 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
6270 characters (for example, single-byte as opposed to multi-byte characters in
6271 arguments and input files).

6272 **LC_MESSAGES**

6273 Determine the locale that should be used to affect the format and contents of
 6274 diagnostic messages written to standard error.

6275 XSI **NLSPATH** Determine the location of message catalogs for the processing of **LC_MESSAGES**.

6276 **ASYNCHRONOUS EVENTS**

6277 Default.

6278 **STDOUT**

6279 The standard output shall be the text from the input file modified as described in the
 6280 DESCRIPTION section.

6281 **STDERR**

6282 None.

6283 **OUTPUT FILES**

6284 None.

6285 **EXTENDED DESCRIPTION**

6286 None.

6287 **EXIT STATUS**

6288 The following exit values shall be returned:

6289 0 All input files were output successfully.

6290 >0 An error occurred.

6291 **CONSEQUENCES OF ERRORS**

6292 Default.

6293 **APPLICATION USAGE**

6294 None.

6295 **EXAMPLES**

6296 1. The following command:

6297 `asa file`

6298 permits the viewing of *file* (created by a program using FORTRAN-style carriage control
 6299 characters) on a terminal.

6300 2. The following command:

6301 `a.out | asa | lp`

6302 formats the FORTRAN output of **a.out** and directs it to the printer.

6303 **RATIONALE**

6304 The *asa* utility is needed to map “standard” FORTRAN 77 output into a form acceptable to
 6305 contemporary printers. Usually, *asa* is used to pipe data to the *lp* utility; see *lp*.

6306 This utility is generally used only by FORTRAN programs. The standard developers decided to
 6307 retain *asa* to avoid breaking the historical large base of FORTRAN applications that put
 6308 carriage-control characters in their output files. There is no requirement that a system have a
 6309 FORTRAN compiler in order to run applications that need *asa*.

6310 Historical implementations have used an ASCII <form-feed> character in response to a 1 and an
 6311 ASCII <carriage-return> in response to a ‘+’. It is suggested that implementations treat
 6312 characters other than 0, 1, and ‘+’ as <space> in the absence of any compelling reason to do
 6313 otherwise. However, the action is listed here as “unspecified”, permitting an implementation to

6314 provide extensions to access fast multiple-line slewing and channel seeking in a non-portable
6315 manner.

6316 **FUTURE DIRECTIONS**

6317 None.

6318 **SEE ALSO**

6319 *fort77, lp*

6320 **CHANGE HISTORY**

6321 First released in Issue 4.

6322 **Issue 6**

6323 This utility is now marked as part of the FORTRAN Runtime Utilities option. |

6324 The normative text is reworded to avoid use of the term “must” for application requirements. |

6325 NAME

6326 at — execute commands at a later time

6327 SYNOPSIS

6328 UP at [-m][-f file][-q queueName] -t time_arg

6329 at [-m][-f file][-q queueName] timespec ...

6330 at -r at_job_id ...

6331 at -l -q queueName

6332 at -l [at_job_id ...]

6333

6334 DESCRIPTION

6335 The *at* utility shall read commands from standard input and group them together as an *at-job*, to
6336 be executed at a later time.6337 The *at-job* shall be executed in a separate invocation of the shell, running in a separate process
6338 group with no controlling terminal, except that the environment variables, current working
6339 directory, file creation mask, and other implementation-dependent execution-time attributes in
6340 effect when the *at* utility is executed shall be retained and used when the *at-job* is executed.6341 When the *at-job* is submitted, the *at_job_id* and scheduled time shall be written to standard error.
6342 The *at_job_id* is an identifier that shall be a string consisting solely of alphanumeric characters
6343 and the period character. The *at_job_id* shall be assigned by the system when the job is scheduled
6344 such that it uniquely identifies a particular job.6345 User notification and the processing of the job's standard output and standard error are
6346 described under the *-m* option.6347 XSI Users are permitted to use *at* if their name appears in the file */usr/lib/cron/at.allow*. If that file
6348 does not exist, the file */usr/lib/cron/at.deny* is checked to determine whether the user should be
6349 denied access to *at*. If neither file exists, only a process with the appropriate privileges is
6350 allowed to submit a job. If only *at.deny* exists and is empty, global usage is permitted. The
6351 *at.allow* and *at.deny* files consist of one user name per line.

6352 OPTIONS

6353 The *at* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,
6354 Section 12.2, Utility Syntax Guidelines.

6355 The following options shall be supported:

6356 *-f file* Specify the path name of a file to be used as the source of the *at-job*, instead of
6357 standard input.6358 *-l* (The letter ell.) Report all jobs scheduled for the invoking user if no *at_job_id*
6359 operands are specified. If *at_job_ids* are specified, report only information for these
6360 jobs. The output shall be written to standard output.6361 *-m* Send mail to the invoking user after the *at-job* has run, announcing its completion.
6362 Standard output and standard error produced by the *at-job* shall be mailed to the
6363 user as well, unless redirected elsewhere. Mail shall be sent even if the job
6364 produces no output.6365 If *-m* is not used, the job's standard output and standard error shall be provided to
6366 MAN the user by means of mail, unless they are redirected elsewhere; if there is no such
6367 output to provide, the implementation need not notify the user of the job's
6368 completion.

6369	-q <i>queuename</i>	
6370		Specify in which queue to schedule a job for submission. When used with the -l
6371		option, limit the search to that particular queue. By default, at-jobs shall be
6372		scheduled in queue <i>a</i> . In contrast, queue <i>b</i> shall be reserved for batch jobs; see
6373		<i>batch</i> . The meanings of all other <i>queuenames</i> are implementation-dependent. If -q
6374		is specified along with either of the -t <i>time_arg</i> or <i>timespec</i> arguments, the results
6375		are unspecified.
6376	-r	Remove the jobs with the specified <i>at_job_id</i> operands that were previously
6377		scheduled by the <i>at</i> utility.
6378	-t <i>time_arg</i>	Submit the job to be run at the time specified by the <i>time</i> option-argument, which
6379		the application shall ensure has the format as specified by the <i>touch -t time</i> utility.

6380 OPERANDS

6381 The following operands shall be supported:

6382	<i>at_job_id</i>	The name reported by a previous invocation of the <i>at</i> utility at the time the job was
6383		scheduled.
6384	<i>timespec</i>	Submit the job to be run at the date and time specified. All of the <i>timespec</i> operands
6385		are interpreted as if they were separated by <space> characters and concatenated,
6386		and shall be parsed as described in the grammar at the end of this section. The date
6387		and time shall be interpreted as being in the timezone of the user (as determined
6388		by the <i>TZ</i> variable), unless a timezone name appears as part of <i>time</i> , below.
6389		In the POSIX locale, the following describes the three parts of the time
6390		specification string. All of the values from the <i>LC_TIME</i> categories in the POSIX
6391		locale shall be recognized in a case-insensitive manner.
6392	<i>time</i>	The time can be specified as one, two, or four digits. One-digit and
6393		two-digit numbers shall be taken to be hours; four-digit numbers to
6394		be hours and minutes. The time can alternatively be specified as two
6395		numbers separated by a colon, meaning <i>hour:minute</i> . An AM/PM
6396		indication (one of the values from the am_pm keywords in the
6397		<i>LC_TIME</i> locale category) can follow the time; otherwise, a 24-hour
6398		clock time shall be understood. A timezone name can also follow to
6399		further qualify the time. The acceptable timezone names are
6400		implementation-dependent, except that they shall be case-insensitive
6401		and the string utc is supported to indicate the time is in Coordinated
6402		Universal Time. In the POSIX locale, the <i>time</i> field can also be one of
6403		the following tokens:
6404	midnight	Indicates the time 12:00 am (00:00).
6405	noon	Indicates the time 12:00 pm.
6406	now	Indicates the current day and time. Invoking <i>at <now></i>
6407		shall submit an at-job for potentially immediate
6408		execution (that is, subject only to unspecified
6409		scheduling delays).
6410	<i>date</i>	An optional <i>date</i> can be specified as either a month name (one of the
6411		values from the mon or abmon keywords in the <i>LC_TIME</i> locale
6412		category) followed by a day number (and possibly year number
6413		preceded by a comma), or a day of the week (one of the values from
6414		the day or abday keywords in the <i>LC_TIME</i> locale category). In the
6415		POSIX locale, two special days shall be recognized:

6416 **today** Indicates the current day.

6417 **tomorrow** Indicates the day following the current day.

6418 If no *date* is given, **today** shall be assumed if the given time is greater
6419 than the current time, and **tomorrow** shall be assumed if it is less. If
6420 the given month is less than the current month (and no year is given),
6421 next year shall be assumed.

6422 *increment* The optional *increment* shall be a number preceded by a plus sign
6423 ('+') and suffixed by one of the following: **minutes, hours, days,**
6424 **weeks, months, or years.** (The singular forms shall be also
6425 accepted.) The keyword **next** shall be equivalent to an increment
6426 number of +1. For example, the following are equivalent commands:

6427 at 2pm + 1 week
6428 at 2pm next week

6429 The following grammar describes the precise format of *timespec* in the POSIX locale. The general
6430 conventions for this style of grammar are described in Section 1.10 on page 24. This formal
6431 syntax shall take precedence over the preceding text syntax description. The longest possible
6432 token or delimiter shall be recognized at a given point. When used in a *timespec*, white space
6433 shall also delimit tokens.

6434 %token hr24clock_hr_min
6435 %token hr24clock_hour
6436 /*
6437 A hr24clock_hr_min is a one, two, or four-digit number. A one-digit
6438 or two-digit number constitutes a hr24clock_hour. A hr24clock_hour
6439 may be any of the single digits 0-9, or may be double digits, ranging
6440 from 00-23. If a hr24clock_hr_min is a four digit number, the
6441 first two digits shall be a valid hr24clock_hour, while the last two
6442 represent the number of minutes, from 00-59.
6443 */

6444 %token wallclock_hr_min
6445 %token wallclock_hour
6446 /*
6447 A wallclock_hr_min is a one, two-digit, or four-digit number.
6448 A one-digit or two-digit number constitutes a wallclock_hour.
6449 A wallclock_hour may be any of the single digits 1-9, or may
6450 be double digits, ranging from 01-12. If a wallclock_hr_min
6451 is a four-digit number, the first two digits shall be a valid
6452 wallclock_hour, while the last two represent the number of
6453 minutes, from 00-59.
6454 */

6455 %token minute
6456 /*
6457 A minute is a one or two-digit number whose values can be 0-9
6458 or 00-59.
6459 */

6460 %token day_number
6461 /*
6462 A day_number is a number in the range appropriate for the particular
6463 month and year specified by month_name and year_number, respectively.


```

6464         If no year_number is given, the current year is assumed if the given
6465         date and time are later this year. If no year_number is given and
6466         the date and time have already occurred this year and the month is
6467         not the current month, next year is the assumed year.
6468     */

6469     %token year_number
6470     /*
6471         A year_number is a four-digit number representing the year A.D., in
6472         which the at_job is to be run.
6473     */

6474     %token inc_number
6475     /*
6476         The inc_number is the number of times the succeeding increment
6477         period is to be added to the specified date and time.
6478     */

6479     %token timezone_name
6480     /*
6481         The name of an optional timezone suffix to the time field, in an
6482         implementation-dependent format.
6483     */

6484     %token month_name
6485     /*
6486         One of the values from the mon or abmon keywords in the LC_TIME
6487         locale category.
6488     */

6489     %token day_of_week
6490     /*
6491         One of the values from the day or abday keywords in the LC_TIME
6492         locale category.
6493     */

6494     %token am_pm
6495     /*
6496         One of the values from the am_pm keyword in the LC_TIME locale
6497         category.
6498     */

6499     %start timespec
6500     %%
6501     timespec      : time
6502                   | time date
6503                   | time increment
6504                   | time date increment
6505                   | nowspec
6506                   ;

6507     nowspec       : "now"
6508                   | "now" increment
6509                   ;

6510     time          : hr24clock_hr_min
6511                   | hr24clock_hr_min timezone_name

```

```

6512         | hr24clock_hour ":" minute
6513         | hr24clock_hour ":" minute timezone_name
6514         | wallclock_hr_min am_pm
6515         | wallclock_hr_min am_pm timezone_name
6516         | wallclock_hour ":" minute am_pm
6517         | wallclock_hour ":" minute am_pm timezone_name
6518         | "noon"
6519         | "midnight"
6520         ;

6521     date      : month_name day_number
6522         | month_name day_number "," year_number
6523         | day_of_week
6524         | "today"
6525         | "tomorrow"
6526         ;

6527     increment : "+" inc_number inc_period
6528         | "next" inc_period
6529         ;

6530     inc_period : "minute" | "minutes"
6531         | "hour" | "hours"
6532         | "day" | "days"
6533         | "week" | "weeks"
6534         | "month" | "months"
6535         | "year" | "years"
6536         ;

```

6537 STDIN

6538 The standard input shall be a text file consisting of commands acceptable to the shell command
 6539 language described in Chapter 2 on page 35. The standard input shall only be used if no **-f file**
 6540 option is specified.

6541 INPUT FILES

6542 See the STDIN section.

6543 XSI The text files **/usr/lib/cron/at.allow** and **/usr/lib/cron/at.deny** contain user names, one per line, of
 6544 users who are, respectively, authorized or denied access to the *at* and *batch* utilities.

6545 ENVIRONMENT VARIABLES

6546 The following environment variables shall affect the execution of *at*:

6547 **LANG** Provide a default value for the internationalization variables that are unset or null.
 6548 If **LANG** is unset or null, the corresponding value from the implementation-
 6549 dependent default locale shall be used. If any of the internationalization variables
 6550 contains an invalid setting, the utility shall behave as if none of the variables had
 6551 been defined.

6552 **LC_ALL** If set to a non-empty string value, override the values of all the other
 6553 internationalization variables.

6554 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as
 6555 characters (for example, single-byte as opposed to multi-byte characters in
 6556 arguments and input files).

6557 **LC_MESSAGES**

6558 Determine the locale that should be used to affect the format and contents of

6559 diagnostic messages written to standard error and informative messages written to
6560 standard output.

6561 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC_MESSAGES*.

6562 **LC_TIME** Determine the format and contents for date and time strings written and accepted
6563 by *at*.

6564 **SHELL** Determine a name of a command interpreter to be used to invoke the at-job. If the
6565 variable is unset or null, *sh* shall be used. If it is set to a value other than a name for
6566 *sh*, the implementation shall do one of the following: use that shell; use *sh*; use the
6567 login shell from the user database; or any of the preceding accompanied by a
6568 warning diagnostic about which was chosen.

6569 **TZ** Determine the timezone. The job shall be submitted for execution at the time
6570 specified by *timespec* or *-t time* relative to the timezone specified by the *TZ*
6571 variable. If *timespec* specifies a timezone, it shall override *TZ*. If *timespec* does not
6572 specify a timezone and *TZ* is unset or null, an unspecified default timezone shall
6573 be used.

6574 **ASYNCHRONOUS EVENTS**

6575 Default.

6576 **STDOUT**

6577 When standard input is a terminal, prompts of unspecified format for each line of the user input
6578 described in the STDIN section may be written to standard output.

6579 In the POSIX locale, the following shall be written to the standard output for each job when jobs
6580 are listed in response to the *-l* option:

6581 "%s\t%s\n", *at_job_id*, <*date*>

6582 where *date* shall be equivalent in format to the output of:

6583 date +"%a %b %e %T %Y"

6584 The date and time written shall be adjusted so that they appear in the timezone of the user (as
6585 determined by the *TZ* variable).

6586 **STDERR**

6587 In the POSIX locale, the following shall be written to standard error when a job has been
6588 successfully submitted:

6589 "job %s at %s\n", *at_job_id*, <*date*>

6590 where *date* has the same format as is described in the STDOUT section. Neither this, nor warning
6591 messages concerning the selection of the command interpreter, shall be considered a diagnostic
6592 that changes the exit status.

6593 Diagnostic messages, if any, shall be written to standard error.

6594 **OUTPUT FILES**

6595 None.

6596 **EXTENDED DESCRIPTION**

6597 None.

6598 **EXIT STATUS**

6599 The following exit values shall be returned:

6600 0 The *at* utility successfully submitted, removed, or listed a job or jobs.

6601 >0 An error occurred.

6602 CONSEQUENCES OF ERRORS

6603 The job shall not be scheduled, removed, or listed.

6604 APPLICATION USAGE

6605 The format of the *at* command line shown here is guaranteed only for the POSIX locale. Other
6606 cultures may be supported with substantially different interfaces, although implementations are
6607 encouraged to provide comparable levels of functionality.

6608 Since the commands run in a separate shell invocation, running in a separate process group with
6609 no controlling terminal, open file descriptors, traps, and priority inherited from the invoking
6610 environment are lost.

6611 Some implementations do not allow substitution of different shells using *SHELL*. System V
6612 systems, for example, have used the login shell value for the user in */etc/passwd*. To select
6613 reliably another command interpreter, the user must include it as part of the script, such as:

```
6614           $ at 1800
6615           myshell myscript
6616           job ... at ...
6617           $
```

6618 Application writers should note that this utility need not be provided on systems that do not
6619 support the User Portability Utilities option.

6620 EXAMPLES

6621 1. This sequence can be used at a terminal:

```
6622           at -m 0730 tomorrow
6623           sort < file >outfile
6624           EOT
```

6625 2. This sequence, which demonstrates redirecting standard error to a pipe, is useful in a
6626 command procedure (the sequence of output redirection specifications is significant):

```
6627           at now + 1 hour <<!
6628           diff file1 file2 2>&1 >outfile | mailx mygroup
6629           !
```

6630 3. To have a job reschedule itself, *at* can be invoked from within the at-job. For example, this
6631 daily processing script named **my.daily** runs every day (although *crontab* is a more
6632 appropriate vehicle for such work):

```
6633           # my.daily runs every day
6634           daily processing
6635           at now tomorrow < my.daily
```

6636 4. The spacing of the three portions of the POSIX locale *timespec* is quite flexible as long as
6637 there are no ambiguities. Examples of various times and operand presentation include:

```
6638           at 0815am Jan 24
6639           at 8 :15amjan24
6640           at now "+ 1day"
6641           at 5 pm FRIday
6642           at '17
6643           utc+
6644           30minutes'
```

RATIONALE

The *at* utility reads from standard input the commands to be executed at a later time. It may be useful to redirect standard output and standard error within the specified commands.

The *-t time* option was added as a new capability to support an internationalized way of specifying a time for execution of the submitted job.

Early proposals added a “jobname” concept as a way of giving submitted jobs names that are meaningful to the user submitting them. The historical, system-specified *at_job_id* gives no indication of what the job is. Upon further reflection, it was decided that the benefit of this was not worth the change in historical interface. It is anticipated that considerably more sophisticated ways of controlling background or batch work will be the subject of a future version of this volume of IEEE Std. 1003.1-200x.

The *-q* option historically has been an undocumented option, used mainly by the *batch* utility.

The System V *-m* option was added to provide a method for informing users that an at-job had completed. Otherwise, users are only informed when output to standard error or standard output are not redirected.

The behavior of *at <now>* was changed in an early proposal from being unspecified to submitting a job for potentially immediate execution. Historical BSD *at* implementations support this. Historical System V implementations give an error in that case, but a change to the System V versions should have no backwards compatibility ramifications.

On BSD-based systems, a *-u user* option has allowed those with appropriate privileges to access the work of other users. Since this is primarily a system administration feature and is not universally implemented, it has been omitted. Similarly, a specification for the output format for user with appropriate privileges viewing the queues of other users has been omitted.

The *-f file* option from System V is used instead of the BSD method of using the last operand as the path name. The BSD method is ambiguous—does:

```
at 1200 friday
```

mean the same thing if there is a file named **friday** in the current directory?

The *at_job_id* is composed of a limited character set in historical practice, and it is mandated here to invalidate systems that might try using characters that require shell quoting or that could not be easily parsed by shell scripts.

The *at* utility varies between System V and BSD systems in the way timezones are used. On System V systems, the *TZ* variable affects the at-job submission times and the times displayed for the user. On BSD systems, *TZ* is not taken into account. The BSD behavior is easily achieved with the current specification. If the user wishes to have the timezone default to that of the system, they merely need to issue the *at* command immediately following an unsetting or null assignment to *TZ*. For example:

```
TZ= at noon ...
```

gives the desired BSD result.

While the yacc-like grammar specified in the OPERANDS section is lexically unambiguous with respect to the digit strings, a lexical analyzer would probably be written to look for and return digit strings in those cases. The parser could then check whether the digit string returned is a valid *day_number*, *year_number*, and so on, based on the context.

6687 **FUTURE DIRECTIONS**

6688 None.

6689 **SEE ALSO**6690 *batch, crontab*6691 **CHANGE HISTORY**

6692 First released in Issue 2.

6693 **Issue 4**

6694 Aligned with the ISO/IEC 9945-2: 1993 standard.

6695 **Issue 6**

6696 This utility is now marked as part of the User Portability Utilities option.

6697 The following new requirements on POSIX implementations derive from alignment with the
 6698 Single UNIX Specification:

- 6699 • If **-m** is not used, the job's standard output and standard error are provided to the user by
 6700 mail.

6701 The effects of using the **-q** and **-t** options as defined in the IEEE P1003.2b draft standard are
 6702 specified.

6703 The normative text is reworded to avoid use of the term “must” for application requirements.

6704 **NAME**6705 *awk* — pattern scanning and processing language6706 **SYNOPSIS**6707 *awk* [-F *ERE*][-v *assignment*] ... *program* [*argument* ...]6708 *awk* [-F *ERE*] -f *progfile* ... [-v *assignment*] ... [*argument* ...]6709 **DESCRIPTION**

6710 The *awk* utility shall execute programs written in the *awk* programming language, which is
 6711 specialized for textual data manipulation. An *awk* program is a sequence of patterns and
 6712 corresponding actions. When input is read that matches a pattern, the action associated with
 6713 that pattern is carried out.

6714 Input shall be interpreted as a sequence of records. By default, a record is a line, but this can be
 6715 changed by using the **RS** built-in variable. Each record of input shall be matched in turn against
 6716 each pattern in the program. For each pattern matched, the associated action shall be executed.

6717 The *awk* utility shall interpret each input record as a sequence of fields where, by default, a field
 6718 is a string of non-<blank> characters. This default white-space field delimiter can be changed by
 6719 using the **FS** built-in variable or the -F *ERE*. The *awk* utility shall denote the first field in a
 6720 record \$1, the second \$2, and so on. The symbol \$0 shall refer to the entire record; setting any
 6721 other field causes the re-evaluation of \$0. Assigning to \$0 shall reset the values of all other fields
 6722 and the **NF** built-in variable.

6723 **OPTIONS**

6724 The *awk* utility shall conform to the System Interface Definitions volume of
 6725 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

6726 The following options shall be supported:

6727 -F *ERE* Define the input field separator to be the extended regular expression *ERE*, before
 6728 any input is read; see **Regular Expressions** on page 195.

6729 -f *progfile* Specify the path name of the file *progfile* containing an *awk* program. If multiple
 6730 instances of this option are specified, the concatenation of the files specified as
 6731 *progfile* in the order specified shall be the *awk* program. The *awk* program can
 6732 alternatively be specified in the command line as a single argument.

6733 -v *assignment* The application shall ensure that the *assignment* argument is in the same form as an
 6734 *assignment* operand. The specified variable assignment shall occur prior to
 6735 executing the *awk* program, including the actions associated with **BEGIN** patterns
 6736 (if any). Multiple occurrences of this option can be specified.
 6737

6738 **OPERANDS**

6739 The following operands shall be supported:

6740 *program* If no -f option is specified, the first operand to *awk* shall be the text of the *awk*
 6741 program. The application shall supply the *program* operand as a single argument to
 6742 *awk*. If the text does not end in a <newline> character, *awk* shall interpret the text
 6743 as if it did.

6744 *argument* Either of the following two types of *argument* can be intermixed:

6745 *file* A path name of a file that contains the input to be read, which is
 6746 matched against the set of patterns in the program. If no *file* operands
 6747 are specified, or if a *file* operand is '-', the standard input shall be
 6748 used.

6749 *assignment* An operand that begins with an underscore or alphabetic character
 6750 from the portable character set (see the table in the System Interface
 6751 Definitions volume of IEEE Std. 1003.1-200x, Section 6.1, Portable
 6752 Character Set), followed by a sequence of underscores, digits, and
 6753 alphabets from the portable character set, followed by the '='
 6754 character, shall specify a variable assignment rather than a path
 6755 name. The characters before the '=' represent the name of an *awk*
 6756 variable; if that name is an *awk* reserved word (see **Grammar** on page
 6757 204) the behavior is undefined. The characters following the equal
 6758 sign shall be interpreted as if they appeared in the *awk* program
 6759 preceded and followed by a double-quote ('"') character, as a
 6760 **STRING** token (see **Grammar** on page 204), except that if the last
 6761 character is an unescaped backslash, it shall be interpreted as a literal
 6762 backslash rather than as the first character of the sequence "\".
 6763 The variable shall be assigned the value of that **STRING** token and,
 6764 if appropriate, shall be considered a *numeric string* (see **Expressions**
 6765 **in awk** on page 190), the variable shall also be assigned its numeric
 6766 value. Each such variable assignment shall occur just prior to the
 6767 processing of the following *file*, if any. Thus, an assignment before
 6768 the first *file* argument shall be executed after the **BEGIN** actions (if
 6769 any), while an assignment after the last *file* argument shall occur
 6770 before the **END** actions (if any). If there are no *file* arguments,
 6771 assignments shall be executed before processing the standard input.

6772 **STDIN**

6773 The standard input shall be used only if no *file* operands are specified, or if a *file* operand is '-';
 6774 see the INPUT FILES section. If the *awk* program contains no actions and no patterns, but is
 6775 otherwise a valid *awk* program, standard input and any *file* operands shall not be read and *awk*
 6776 shall exit with a return status of zero.

6777 **INPUT FILES**

6778 Input files to the *awk* program from any of the following sources shall be text files:

- 6779 • Any *file* operands or their equivalents, achieved by modifying the *awk* variables **ARGV** and
- 6780 **ARGC**
- 6781 • Standard input in the absence of any *file* operands
- 6782 • Arguments to the **getline** function

6783 Whether the variable **RS** is set to a value other than a <newline> character or not, for these files,
 6784 implementations shall support records terminated with the specified separator up to
 6785 {**LINE_MAX**} bytes and may support longer records.

6786 If **-f progfile** is specified, the application shall ensure that the files named by each of the *progfile*
 6787 option-arguments are text files containing an *awk* program.

6788 **ENVIRONMENT VARIABLES**

6789 The following environment variables shall affect the execution of *awk*:

6790 **LANG** Provide a default value for the internationalization variables that are unset or null.
 6791 If **LANG** is unset or null, the corresponding value from the implementation-
 6792 dependent default locale shall be used. If any of the internationalization variables
 6793 contains an invalid setting, the utility shall behave as if none of the variables had
 6794 been defined.

6795 **LC_ALL** If set to a non-empty string value, override the values of all the other
6796 internationalization variables.

6797 **LC_COLLATE**
6798 Determine the locale for the behavior of ranges, equivalence classes, and multi-
6799 character collating elements within regular expressions and in comparisons of
6800 string values.

6801 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as
6802 characters (for example, single-byte as opposed to multi-byte characters in
6803 arguments and input files), the behavior of character classes within regular
6804 expressions, the identification of characters as letters, and the mapping of
6805 uppercase and lowercase characters for the **toupper** and **tolower** functions.

6806 **LC_MESSAGES**
6807 Determine the locale that should be used to affect the format and contents of
6808 diagnostic messages written to standard error.

6809 **LC_NUMERIC**
6810 Determine the radix character used when interpreting numeric input, performing
6811 conversions between numeric and string values, and formatting numeric output.
6812 Regardless of locale, the period character (the decimal-point character of the
6813 POSIX locale) is the decimal-point character recognized in processing *awk*
6814 programs (including assignments in command line arguments).

6815 XSI **NLSPATH** Determine the location of message catalogs for the processing of **LC_MESSAGES**.

6816 **PATH** Determine the search path when looking for commands executed by *system(expr)*,
6817 or input and output pipes; see the System Interface Definitions volume of
6818 IEEE Std. 1003.1-200x, Chapter 8, Environment Variables.

6819 In addition, all environment variables shall be visible via the *awk* variable **ENVIRON**.

6820 **ASYNCHRONOUS EVENTS**
6821 Default.

6822 **STDOUT**
6823 The nature of the output files depends on the *awk* program.

6824 **STDERR**
6825 Used only for diagnostic messages.

6826 **OUTPUT FILES**
6827 The nature of the output files depends on the *awk* program.

6828 **EXTENDED DESCRIPTION**

6829 **Overall Program Structure**
6830 An *awk* program is composed of pairs of the form:
6831 *pattern { action }*
6832 Either the pattern or the action (including the enclosing brace characters) can be omitted.
6833 A missing pattern shall match any record of input, and a missing action shall be equivalent to:
6834 *{ print }*
6835 Execution of the *awk* program shall start by first executing the actions associated with all **BEGIN**
6836 patterns in the order they occur in the program. Then each *file* operand (or standard input if no

files were specified) shall be processed in turn by reading data from the file until a record separator is seen (<newline> by default). Before the first reference to a field in the record is evaluated, the record shall be split into fields, according to the rules in **Regular Expressions** on page 195, using the value of **FS** that was current at the time the record was read. Each pattern in the program then shall be evaluated in the order of occurrence, and the action associated with each pattern that matches the current record executed. The action for a matching pattern shall be executed before evaluating subsequent patterns. Finally, the actions associated with all **END** patterns shall be executed in the order they occur in the program.

Expressions in awk

Expressions describe computations used in *patterns* and *actions*. In the following table, valid expression operations are given in groups from highest precedence first to lowest precedence last, with equal-precedence operators grouped between horizontal lines. In expression evaluation, where the grammar is formally ambiguous, higher precedence operators shall be evaluated before lower precedence operators. In this table *expr*, *expr1*, *expr2*, and *expr3* represent any expression, while *lvalue* represents any entity that can be assigned to (that is, on the left side of an assignment operator). The precise syntax of expressions is given in **Grammar** on page 204.

Table 4-1 Expressions in Decreasing Precedence in *awk*

Syntax	Name	Type of Result	Associativity
(<i>expr</i>)	Grouping	Type of <i>expr</i>	N/A
<i>\$expr</i>	Field reference	String	N/A
++ <i>lvalue</i>	Pre-increment	Numeric	N/A
-- <i>lvalue</i>	Pre-decrement	Numeric	N/A
<i>lvalue</i> ++	Post-increment	Numeric	N/A
<i>lvalue</i> --	Post-decrement	Numeric	N/A
<i>expr</i> ^ <i>expr</i>	Exponentiation	Numeric	Right
! <i>expr</i>	Logical not	Numeric	N/A
+ <i>expr</i>	Unary plus	Numeric	N/A
- <i>expr</i>	Unary minus	Numeric	N/A
<i>expr</i> * <i>expr</i>	Multiplication	Numeric	Left
<i>expr</i> / <i>expr</i>	Division	Numeric	Left
<i>expr</i> % <i>expr</i>	Modulus	Numeric	Left
<i>expr</i> + <i>expr</i>	Addition	Numeric	Left
<i>expr</i> - <i>expr</i>	Subtraction	Numeric	Left
<i>expr</i> <i>expr</i>	String concatenation	String	Left
<i>expr</i> < <i>expr</i>	Less than	Numeric	None
<i>expr</i> <= <i>expr</i>	Less than or equal to	Numeric	None
<i>expr</i> != <i>expr</i>	Not equal to	Numeric	None
<i>expr</i> == <i>expr</i>	Equal to	Numeric	None
<i>expr</i> > <i>expr</i>	Greater than	Numeric	None
<i>expr</i> >= <i>expr</i>	Greater than or equal to	Numeric	None
<i>expr</i> ~ <i>expr</i>	ERE match	Numeric	None
<i>expr</i> !~ <i>expr</i>	ERE non-match	Numeric	None
<i>expr</i> in array	Array membership	Numeric	Left

Syntax	Name	Type of Result	Associativity
(<i>index</i>) in <i>array</i>	Multi-dimension array membership	Numeric	Left
<i>expr</i> && <i>expr</i>	Logical AND	Numeric	Left
<i>expr</i> <i>expr</i>	Logical OR	Numeric	Left
<i>expr1</i> ? <i>expr2</i> : <i>expr3</i>	Conditional expression	Type of selected <i>expr2</i> or <i>expr3</i>	Right
<i>lvalue</i> ^= <i>expr</i>	Exponentiation assignment	Numeric	Right
<i>lvalue</i> %= <i>expr</i>	Modulus assignment	Numeric	Right
<i>lvalue</i> *= <i>expr</i>	Multiplication assignment	Numeric	Right
<i>lvalue</i> /= <i>expr</i>	Division assignment	Numeric	Right
<i>lvalue</i> += <i>expr</i>	Addition assignment	Numeric	Right
<i>lvalue</i> -= <i>expr</i>	Subtraction assignment	Numeric	Right
<i>lvalue</i> = <i>expr</i>	Assignment	Type of <i>expr</i>	Right

Each expression shall have either a string value, a numeric value, or both. Except as stated for specific contexts, the value of an expression shall be implicitly converted to the type needed for the context in which it is used. A string value shall be converted to a numeric value by the equivalent of the following calls to functions defined by the ISO C standard:

```
setlocale(LC_NUMERIC, "");
numeric_value = atof(string_value);
```

A numeric value that is exactly equal to the value of an integer shall be converted to a string by the equivalent of a call to the **sprintf** function (see **String Functions** on page 201) with the string "%d" as the *fmt* argument and the numeric value being converted as the first and only *expr* argument. Any other numeric value shall be converted to a string by the equivalent of a call to the **sprintf** function with the value of the variable **CONVFMT** as the *fmt* argument and the numeric value being converted as the first and only *expr* argument. The result of the conversion is unspecified if the value of **CONVFMT** is not a floating-point format specification. This volume of IEEE Std. 1003.1-200x specifies no explicit conversions between numbers and strings. An application can force an expression to be treated as a number by adding zero to it, or can force it to be treated as a string by concatenating the null string (" ") to it.

A string value shall be considered a *numeric string* if it comes from one of the following:

1. Field variables
2. Input from the *getline()* function
3. **FILENAME**
4. **ARGV** array elements
5. **ENVIRON** array elements
6. Array elements created by the *split()* function
7. A command line variable assignment
8. Variable assignment from another numeric string variable

and after all the following conversions have been applied, the resulting string would lexically be recognized as a **NUMBER** token as described by the lexical conventions in **Grammar** on page 204:

- All leading and trailing <blank>s are discarded

6925 • If the first non-<blank> character is '+' or '-', it is discarded |

6926 • Changing each occurrence of the decimal point character from the current locale to a period |

6927 If a '-' character is ignored in the preceding description, the numeric value of the *numeric string* |

6928 shall be the negation of the numeric value of the recognized **NUMBER** token. Otherwise, the |

6929 numeric value of the *numeric string* shall be the numeric value of the recognized **NUMBER** |

6930 token. Whether or not a string is a *numeric string* shall be relevant only in contexts where that |

6931 term is used in this section.

6932 When an expression is used in a Boolean context, if it has a numeric value, a value of zero shall

6933 be treated as false and any other value shall be treated as true. Otherwise, a string value of the

6934 null string shall be treated as false and any other value shall be treated as true. A Boolean

6935 context shall be one of the following:

6936 • The first subexpression of a conditional expression

6937 • An expression operated on by logical NOT, logical AND, or logical OR

6938 • The second expression of a **for** statement

6939 • The expression of an **if** statement

6940 • The expression of the **while** clause in either a **while** or **do...while** statement

6941 • An expression used as a pattern (as in Overall Program Structure)

6942 All arithmetic shall follow the semantics of floating-point arithmetic as specified by the ISO C

6943 standard.

6944 The value of the expression:

6945 `expr1 ^ expr2`

6946 shall be equivalent to the value returned by the ISO C standard function call:

6947 `pow(expr1, expr2)`

6948 The expression:

6949 `lvalue ^= expr`

6950 shall be equivalent to the ISO C standard expression:

6951 `lvalue = pow(lvalue, expr)`

6952 except that *lvalue* shall be evaluated only once. The value of the expression:

6953 `expr1 % expr2`

6954 shall be equivalent to the value returned by the ISO C standard function call:

6955 `fmod(expr1, expr2)`

6956 The expression:

6957 `lvalue %= expr`

6958 shall be equivalent to the ISO C standard expression:

6959 `lvalue = fmod(lvalue, expr)`

6960 except that *lvalue* shall be evaluated only once.

6961 Variables and fields shall be set by the assignment statement:

6962 *lvalue* = *expression*

6963 and the type of *expression* shall determine the resulting variable type. The assignment includes
 6964 the arithmetic assignments ("+=", "-=", "*=", "/=", "%=", "^=", "++", "--") all of which
 6965 shall produce a numeric result. The left-hand side of an assignment and the target of increment
 6966 and decrement operators can be one of a variable, an array with index, or a field selector.

6967 The *awk* language supplies arrays that are used for storing numbers or strings. Arrays need not
 6968 be declared. They shall initially be empty, and their sizes shall change dynamically. The
 6969 subscripts, or element identifiers, are strings, providing a type of associative array capability. An
 6970 array name followed by a subscript within square brackets can be used as an *lvalue* and thus as
 6971 an expression, as described in the grammar; see **Grammar** on page 204. Unsubscripted array
 6972 names can be used in only the following contexts:

- 6973 • A parameter in a function definition or function call
- 6974 • The **NAME** token following any use of the keyword **in** as specified in the grammar (see
 6975 **Grammar** on page 204); if the name used in this context is not an array name, the behavior is
 6976 undefined

6977 A valid array *index* shall consist of one or more comma-separated expressions, similar to the way
 6978 in which multi-dimensional arrays are indexed in some programming languages. Because *awk*
 6979 arrays are really one-dimensional, such a comma-separated list shall be converted to a single
 6980 string by concatenating the string values of the separate expressions, each separated from the
 6981 other by the value of the **SUBSEP** variable. Thus, the following two index operations shall be
 6982 equivalent:

6983 *var*[*expr1*, *expr2*, ... *exprn*]

6984 *var*[*expr1* **SUBSEP** *expr2* **SUBSEP** ... **SUBSEP** *exprn*]

6985 The application shall ensure that a multi-dimensioned *index* used with the **in** operator is
 6986 parenthesized. The **in** operator, which tests for the existence of a particular array element, shall
 6987 not cause that element to exist. Any other reference to a nonexistent array element shall
 6988 automatically create it.

6989 Comparisons (with the '<', '<=', '!=', '==', '>', and '>=' operators) shall be made
 6990 numerically if both operands are numeric, if one is numeric and the other has a string value that
 6991 is a numeric string, or if one is numeric and the other has the uninitialized value. Otherwise,
 6992 operands shall be converted to strings as required and a string comparison shall be made using
 6993 the locale-specific collation sequence. The value of the comparison expression shall be 1 if the
 6994 relation is true, or 0 if the relation is false.

6995 Variables and Special Variables

6996 Variables can be used in an *awk* program by referencing them. With the exception of function
 6997 parameters (see **User-Defined Functions** on page 203), they are not explicitly declared. Function
 6998 parameter names shall be local to the function; all other variable names shall be global. The same
 6999 name shall not be used as both a function parameter name and as the name of a function or a
 7000 special *awk* variable. The same name shall not be used both as a variable name with global scope
 7001 and as the name of a function. The same name shall not be used within the same scope both as a
 7002 scalar variable and as an array. Uninitialized variables, including scalar variables, array
 7003 elements, and field variables, shall have an uninitialized value. An uninitialized value shall have
 7004 both a numeric value of zero and a string value of the empty string. Evaluation of variables with
 7005 an uninitialized value, to either string or numeric, shall be determined by the context in which
 7006 they are used.

Field variables shall be designated by a '\$' followed by a number or numerical expression. The effect of the field number *expression* evaluating to anything other than a non-negative integer is unspecified; uninitialized variables or string values need not be converted to numeric values in this context. New field variables can be created by assigning a value to them. References to nonexistent fields (that is, fields after **\$NF**), shall evaluate to the uninitialized value. Such references shall not create new fields. However, assigning to a nonexistent field (for example, **\$(NF+2)=5**) shall increase the value of **NF**; create any intervening fields with the uninitialized value; and cause the value of **\$0** to be recomputed, with the fields being separated by the value of **OFS**. Each field variable shall have a string value or an uninitialized value when created. Field variables shall have the uninitialized value when created from **\$0** using **FS** and the variable does not contain any characters. If appropriate, the field variable shall be considered a numeric string (see **Expressions in awk** on page 190).

Implementations shall support the following other special variables that are set by *awk*:

- ARGC** The number of elements in the **ARGV** array.
- ARGV** An array of command line arguments, excluding options and the *program* argument, numbered from zero to **ARGC**-1.
- The arguments in **ARGV** can be modified or added to; **ARGC** can be altered. As each input file ends, *awk* shall treat the next non-null element of **ARGV**, up to the current value of **ARGC**-1, inclusive, as the name of the next input file. Thus, setting an element of **ARGV** to null means that it shall not be treated as an input file. The name '-' indicates the standard input. If an argument matches the format of an *assignment* operand, this argument shall be treated as an assignment rather than a *file* argument.
- CONVFMT** The **printf** format for converting numbers to strings (except for output statements, where **OFMT** is used); "% .6g" by default.
- ENVIRON** The variable **ENVIRON** is an array representing the value of the environment, as described in the *exec* functions defined in the System Interfaces volume of IEEE Std. 1003.1-200x. The indices of the array shall be strings consisting of the names of the environment variables, and the value of each array element is a string consisting of the value of that variable. If appropriate, the environment variable shall be considered a *numeric string* (see **Expressions in awk** on page 190), the array element shall also have its numeric value.
- In all cases where the behavior of *awk* is affected by environment variables (including the environment of any commands that *awk* executes via the **system** function or via pipeline redirections with the **print** statement, the **printf** statement, or the **getline** function), the environment used shall be the environment at the time *awk* began executing; it is implementation-dependent whether any modification of **ENVIRON** affects this environment.
- FILENAME** A path name of the current input file. Inside a **BEGIN** action the value is undefined. Inside an **END** action the value is the name of the last input file processed.
- FNR** The ordinal number of the current record in the current file. Inside a **BEGIN** action the value is zero. Inside an **END** action the value is the number of the last record processed in the last file processed.
- FS** Input field separator regular expression; a <space> character by default.
- NF** The number of fields in the current record. Inside a **BEGIN** action, the use of **NF** is undefined unless a **getline** function without a *var* argument is executed

7054		previously. Inside an END action, NF retains the value it had for the last record read, unless a subsequent redirected, getline function without a <i>var</i> argument is performed prior to entering the END action.
7055		
7056		
7057	NR	The ordinal number of the current record from the start of input. Inside a BEGIN action the value is zero. Inside an END action the value is the number of the last record processed.
7058		
7059		
7060	OFMT	The printf format for converting numbers to strings in output statements (see Output Statements on page 199); "% . 6g" by default. The result of the conversion is unspecified if the value of OFMT is not a floating-point format specification.
7061		
7062		
7063	OFS	The print statement output field separation; <space> by default.
7064	ORS	The print statement output record separator; a <newline> character by default.
7065	RLENGTH	The length of the string matched by the match function.
7066	RS	The first character of the string value of RS is the input record separator; a <newline> character by default. If RS contains more than one character, the results are unspecified. If RS is null, then records are separated by sequences of one or more blank lines, leading or trailing blank lines do not result in empty records at the beginning or end of the input, and a <newline> character is always a field separator, no matter what the value of FS is.
7067		
7068		
7069		
7070		
7071		
7072	RSTART	The starting position of the string matched by the match function, numbering from 1. This is always equivalent to the return value of the match function.
7073		
7074	SUBSEP	The subscript separator string for multi-dimensional arrays; the default value is implementation-dependent.
7075		
7076	Regular Expressions	
7077	The <i>awk</i> utility shall make use of the extended regular expression notation (see the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 9.4, Extended Regular Expressions) except that it shall allow the use of C-language conventions for escaping special characters within the EREs, as specified in the table in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 5, File Format Notation ('\\', '\a', '\b', '\f', '\n', '\r', '\t', '\v') and the following table; these escape sequences shall be recognized both inside and outside bracket expressions. Note that records need not be separated by <newline> characters and string constants can contain <newline> characters, so even the "\n" sequence is valid in <i>awk</i> EREs. Using a slash character within an ERE requires the escaping shown in the following table.	
7078		
7079		
7080		
7081		
7082		
7083		
7084		
7085		
7086		

Table 4-2 Escape Sequences in *awk*

Escape Sequence	Description	Meaning
\ "	Backslash quotation-mark	Quotation-mark character
\ /	Backslash slash	Slash character
\ ddd	A backslash character followed by the longest sequence of one, two, or three octal-digit characters (01234567). If all of the digits are 0 (that is, representation of the NUL character), the behavior is undefined.	The character whose encoding is represented by the one, two, or three-digit octal integer. If the size of a byte on the system is greater than nine bits, the valid escape sequence used to represent a byte is implementation-dependent. Multi-byte characters require multiple, concatenated escape sequences of this type, including the leading '\ ' for each byte.
\ c	A backslash character followed by any character not described in this table or in the table in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 5, File Format Notation ('\\', '\a', '\b', '\f', '\n', '\r', '\t', '\v')	Undefined

A regular expression can be matched against a specific field or string by using one of the two regular expression matching operators, '~' and '!~'. These operators shall interpret their right-hand operand as a regular expression and their left-hand operand as a string. If the regular expression matches the string, the '~' expression shall evaluate to a value of 1, and the '!~' expression shall evaluate to a value of 0. (The regular expression matching operation is as defined by the term matched in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 9.1, Regular Expression Definitions, where a match occurs on any part of the string unless the regular expression is limited with the circumflex or dollar sign special characters.) If the regular expression does not match the string, the '~' expression shall evaluate to a value of 0, and the '!~' expression shall evaluate to a value of 1. If the right-hand operand is any expression other than the lexical token **ERE**, the string value of the expression shall be interpreted as an extended regular expression, including the escape conventions described above. Note that these same escape conventions shall also be applied in determining the value of a string literal (the lexical token **STRING**), and thus shall be applied a second time when a string literal is used in this context.

When an **ERE** token appears as an expression in any context other than as the right-hand of the '~' or '!~' operator or as one of the built-in function arguments described below, the value of the resulting expression shall be the equivalent of:

```
$0 ~ /ere/
```

The *ere* argument to the **gsub**, **match**, **sub** functions, and the *fs* argument to the **split** function (see **String Functions** on page 201) shall be interpreted as extended regular expressions. These can be either **ERE** tokens or arbitrary expressions, and shall be interpreted in the same manner as the right-hand side of the '~' or '!~' operator.

An extended regular expression can be used to separate fields by using the **-F ERE** option or by assigning a string containing the expression to the built-in variable **FS**. The default value of the **FS** variable shall be a single <space> character. The following describes **FS** behavior:

1. If **FS** is a null string, the behavior is unspecified.
2. If **FS** is a single character:
 - a. If **FS** is the <space> character, skip leading and trailing <blank> characters; fields shall be delimited by sets of one or more <blank> characters.
 - b. Otherwise, if **FS** is any other character *c*, fields shall be delimited by each single occurrence of *c*.
3. Otherwise, the string value of **FS** shall be considered to be an extended regular expression. Each occurrence of a sequence matching the extended regular expression shall delimit fields.

Except for the `'~'` and `"!~"` operators, and in the **gsub**, **match**, **split**, and **sub** built-in functions, ERE matching shall be based on input records; that is, record separator characters (the first character of the value of the variable **RS**, <newline> by default) cannot be embedded in the expression, and no expression shall match the record separator character. If the record separator is not <newline>, <newline> characters embedded in the expression can be matched. For the `'~'` and `"!~"` operators, and in those four built-in functions, ERE matching shall be based on text strings; that is, any character (including <newline> and the record separator) can be embedded in the pattern, and an appropriate pattern shall match any character. However, in all *awk* ERE matching, the use of one or more NUL characters in the pattern, input record, or text string produces undefined results.

Patterns

A *pattern* is any valid *expression*, a range specified by two expressions separated by comma, or one of the two special patterns **BEGIN** or **END**.

Special Patterns

The *awk* utility shall recognize two special patterns, **BEGIN** and **END**. Each **BEGIN** pattern shall be matched once and its associated action executed before the first record of input is read (except possibly by use of the **getline** function—see **Input/Output and General Functions** on page 202—in a prior **BEGIN** action) and before command line assignment is done. Each **END** pattern shall be matched once and its associated action executed after the last record of input has been read. These two patterns shall have associated actions.

BEGIN and **END** shall not combine with other patterns. Multiple **BEGIN** and **END** patterns shall be allowed. The actions associated with the **BEGIN** patterns shall be executed in the order specified in the program, as are the **END** actions. An **END** pattern can precede a **BEGIN** pattern in a program.

If an *awk* program consists of only actions with the pattern **BEGIN**, and the **BEGIN** action contains no **getline** function, *awk* shall exit without reading its input when the last statement in the last **BEGIN** action is executed. If an *awk* program consists of only actions with the pattern **END** or only actions with the patterns **BEGIN** and **END**, the input shall be read before the statements in the **END** actions are executed.

Expression Patterns

An expression pattern shall be evaluated as if it were an expression in a Boolean context. If the result is true, the pattern shall be considered to match, and the associated action (if any) shall be executed. If the result is false, the action shall not be executed.

Pattern Ranges

A pattern range consists of two expressions separated by a comma; in this case, the action shall be performed for all records between a match of the first expression and the following match of the second expression, inclusive. At this point, the pattern range can be repeated starting at input records subsequent to the end of the matched range.

Actions

An action is a sequence of statements as shown in the grammar in **Grammar** on page 204. Any single statement can be replaced by a statement list enclosed in braces. The application shall ensure that statements in a statement list are separated by <newline> characters or semicolons, and are executed sequentially in the order that they appear.

The *expression* acting as the conditional in an **if** statement shall be evaluated and if it is non-zero or non-null, the following *statement* shall be executed; otherwise, if **else** is present, the statement following the **else** shall be executed.

The **if**, **while**, **do...while**, **for**, **break**, and **continue** statements are based on the ISO C standard, except that the Boolean expressions shall be treated as described in **Expressions in awk** on page 190, and except in the case of:

```
for (variable in array)
```

which shall iterate, assigning each *index* of *array* to *variable* in an unspecified order. The results of adding new elements to *array* within such a **for** loop are undefined. If a **break** or **continue** statement occurs outside of a loop, the behavior is undefined.

The **delete** statement shall remove an individual array element. Thus, the following code deletes an entire array:

```
for (index in array)
    delete array[index]
```

The **next** statement shall cause all further processing of the current input record to be abandoned. The behavior is undefined if a **next** statement appears or is invoked in a **BEGIN** or **END** action.

The **exit** statement shall invoke all **END** actions in the order in which they occur in the program source and then terminate the program without reading further input. An **exit** statement inside an **END** action shall terminate the program without further execution of **END** actions. If an expression is specified in an **exit** statement, its numeric value shall be the exit status of *awk*, unless subsequent errors are encountered or a subsequent **exit** statement with an expression is executed.

Output Statements

Both **print** and **printf** statements shall write to standard output by default. The output shall be written to the location specified by *output_redirection* if one is supplied, as follows:

```
> expression
>> expression
| expression
```

In all cases, the *expression* shall be evaluated to produce a string that is used as a path name into which to write (for '*>*' or '*>>*') or as a command to be executed (for '*|*'). Using the first two forms, if the file of that name is not currently open, it shall be opened, creating it if necessary and using the first form, truncating the file. The output then shall be appended to the file. As long as the file remains open, subsequent calls in which *expression* evaluates to the same string value shall simply append output to the file. The file remains open until the **close** function (see **Input/Output and General Functions** on page 202) is called with an expression that evaluates to the same string value.

The third form shall write output onto a stream piped to the input of a command. The stream shall be created if no stream is currently open with the value of *expression* as its command name. The stream created shall be equivalent to one created by a call to the *popen()* function defined in the System Interfaces volume of IEEE Std. 1003.1-200x with the value of *expression* as the *command* argument and a value of *w* as the *mode* argument. As long as the stream remains open, subsequent calls in which *expression* evaluates to the same string value shall write output to the existing stream. The stream shall remain open until the **close** function (see **Input/Output and General Functions** on page 202) is called with an expression that evaluates to the same string value. At that time, the stream shall be closed as if by a call to the *pclose()* function defined in the System Interfaces volume of IEEE Std. 1003.1-200x.

As described in detail by the grammar in **Grammar** on page 204, these output statements shall take a comma-separated list of *expressions* referred to in the grammar by the non-terminal symbols **expr_list**, **print_expr_list**, or **print_expr_list_opt**. This list is referred to here as the *expression list*, and each member is referred to as an *expression argument*.

The **print** statement shall write the value of each expression argument onto the indicated output stream separated by the current output field separator (see variable **OFS** above), and terminated by the output record separator (see variable **ORS** above). All expression arguments shall be taken as strings, being converted if necessary; this conversion shall be as described in **Expressions in awk** on page 190, with the exception that the **printf** format in **OFMT** shall be used instead of the value in **CONVFMT**. An empty expression list shall stand for the whole input record (\$0).

The **printf** statement shall produce output based on a notation similar to the File Format Notation used to describe file formats in this volume of IEEE Std. 1003.1-200x (see the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 5, File Format Notation). Output shall be produced as specified with the first expression argument as the string *format* and subsequent expression arguments as the strings *arg1* to *argn*, inclusive, with the following exceptions:

1. The *format* shall be an actual character string rather than a graphical representation. Therefore, it cannot contain empty character positions. The *<space>* character in the *format* string, in any context other than a *flag* of a conversion specification, shall be treated as an ordinary character that is copied to the output.
2. If the character set contains a '*Δ*' character and that character appears in the *format* string, it shall be treated as an ordinary character that is copied to the output.

3. The *escape sequences* beginning with a backslash character shall be treated as sequences of ordinary characters that are copied to the output. Note that these same sequences shall be interpreted lexically by *awk* when they appear in literal strings, but they shall not be treated specially by the **printf** statement.
4. A *field width* or *precision* can be specified as the ' * ' character instead of a digit string. In this case the next argument from the expression list shall be fetched and its numeric value taken as the field width or precision.
5. The implementation shall not precede or follow output from the *d* or *u* conversion specifications with <blank> characters not specified by the *format* string.
6. The implementation shall not precede output from the *o* conversion specification with leading zeros not specified by the *format* string.
7. For the *c* conversion specification: if the argument has a numeric value, the character whose encoding is that value shall be output. If the value is zero or is not the encoding of any character in the character set, the behavior is undefined. If the argument does not have a numeric value, the first character of the string value shall be output; if the string does not contain any characters, the behavior is undefined.
8. For each conversion specification that consumes an argument, the next expression argument shall be evaluated. With the exception of the *c* conversion, the value shall be converted (according to the rules specified in **Expressions in awk** on page 190) to the appropriate type for the conversion specification.
9. If there are insufficient expression arguments to satisfy all the conversion specifications in the *format* string, the behavior is undefined.
10. If any character sequence in the *format* string begins with a ' % ' character, but does not form a valid conversion specification, the behavior is unspecified.

Both **print** and **printf** can output at least {LINE_MAX} bytes.

Functions

The *awk* language has a variety of built-in functions: arithmetic, string, input/output, and general.

Arithmetic Functions

The arithmetic functions, except for **int**, shall be based on the ISO C standard. The behavior is undefined in cases where the ISO C standard specifies that an error be returned or that the behavior is undefined. Although the grammar (see **Grammar** on page 204) permits built-in functions to appear with no arguments or parentheses, unless the argument or parentheses are indicated as optional in the following list (by displaying them within the " [] " brackets), such use is undefined.

atan2 (<i>y</i> , <i>x</i>)	Return arctangent of <i>y</i> / <i>x</i> in radians in the range $-\{\pi\}$ to $\{\pi\}$.
cos (<i>x</i>)	Return cosine of <i>x</i> , where <i>x</i> is in radians.
sin (<i>x</i>)	Return sine of <i>x</i> , where <i>x</i> is in radians.
exp (<i>x</i>)	Return the exponential function of <i>x</i> .
log (<i>x</i>)	Return the natural logarithm of <i>x</i> .
sqrt (<i>x</i>)	Return the square root of <i>x</i> .

7299 **int**(*x*) Truncate its argument to an integer. It shall be truncated toward 0 when *x*>0.
 7300 **rand**() Return a random number *n*, such that $0 \leq n < 1$.
 7301 **srand**([*expr*]) Set the seed value for *rand* to *expr* or use the time of day if *expr* is omitted. The
 7302 previous seed value shall be returned.

7303 String Functions

7304 The string functions in the following list shall be supported. Although the grammar (see
 7305 **Grammar** on page 204) permits built-in functions to appear with no arguments or parentheses,
 7306 unless the argument or parentheses are indicated as optional in the following list (by displaying
 7307 them within the " [] " brackets), such use is undefined.

7308 **gsub**(*ere*, *repl* [, *in*])
 7309 Behave like **sub** (see below), except that it shall replace all occurrences of the
 7310 regular expression (like the *ed* utility global substitute) in \$0 or in the *in* argument,
 7311 when specified.

7312 **index**(*s*, *t*) Return the position, in characters, numbering from 1, in string *s* where string *t* first
 7313 occurs, or zero if it does not occur at all.

7314 **length**[(*[s]*)] Return the length, in characters, of its argument taken as a string, or of the whole
 7315 MAN record, \$0, if there is no argument. The use of no argument and no parentheses
 7316 with **length** is obsolescent in the IEEE Std. 1003.1-200x; to be fully portable
 7317 POSIX-conforming applications shall use **length**(\$0) for the length of the whole
 7318 record.

7319 **match**(*s*, *ere*) Return the position, in characters, numbering from 1, in string *s* where the
 7320 extended regular expression *ere* occurs, or zero if it does not occur at all. RSTART
 7321 shall be set to the starting position (which is the same as the returned value), zero
 7322 if no match is found; RLENGTH shall be set to the length of the matched string, -1
 7323 if no match is found.

7324 **split**(*s*, *a* [, *fs*])
 7325 Split the string *s* into array elements *a*[1], *a*[2], ..., *a*[*n*], and return *n*. All elements
 7326 of the array shall be deleted before the split is performed. The separation shall be
 7327 done with the ERE *fs* or with the field separator **FS** if *fs* is not given. Each array
 7328 element shall have a string value when created and, if appropriate, the array
 7329 element shall be considered a numeric string (see **Expressions in awk** on page
 7330 190). The effect of a null string as the value of *fs* is unspecified.

7331 **sprintf**(*fmt*, *expr*, *expr*, ...)
 7332 Format the expressions according to the **printf** format given by *fmt* and return the
 7333 resulting string.

7334 **sub**(*ere*, *repl* [, *in*])
 7335 Substitute the string *repl* in place of the first instance of the extended regular
 7336 expression *ERE* in string *in* and return the number of substitutions. An ampersand
 7337 (' & ') appearing in the string *repl* shall be replaced by the string *from* in that
 7338 matches the ERE. An ampersand preceded with a backslash (' \ ') shall be
 7339 interpreted as the literal ampersand character. Any other occurrence of a backslash
 7340 (for example, preceding any other character) shall be treated as a literal backslash
 7341 character. Note that if *repl* is a string literal (the lexical token **STRING**; see
 7342 **Grammar** on page 204), the handling of the ampersand character occurs after any
 7343 lexical processing, including any lexical backslash escape sequence processing. If
 7344 *in* is specified and it is not an *lvalue* (see **Expressions in awk** on page 190), the

7345 behavior is undefined. If *in* is omitted, *awk* shall use the current record (\$0) in its
7346 place.

7347 **substr**(*s*, *m* [, *n*])
7348 Return the at most *n*-character substring of *s* that begins at position *m*, numbering
7349 from 1. If *n* is missing, or if *n* specifies more characters than are left in the string,
7350 the length of the substring shall be limited by the length of the string *s*.

7351 **tolower**(*s*) Return a string based on the string *s*. Each character in *s* that is an uppercase letter
7352 specified to have a **tolower** mapping by the *LC_CTYPE* category of the current
7353 locale shall be replaced in the returned string by the lowercase letter specified by
7354 the mapping. Other characters in *s* shall be unchanged in the returned string.

7355 **toupper**(*s*) Return a string based on the string *s*. Each character in *s* that is a lowercase letter
7356 specified to have a **toupper** mapping by the *LC_CTYPE* category of the current
7357 locale is replaced in the returned string by the uppercase letter specified by the
7358 mapping. Other characters in *s* are unchanged in the returned string.

7359 All of the preceding functions that take *ERE* as a parameter expect a pattern or a string valued
7360 expression that is a regular expression as defined in **Regular Expressions** on page 195.

7361 Input/Output and General Functions

7362 The input/output and general functions are:

7363 **close**(*expression*)
7364 Close the file or pipe opened by a **print** or **printf** statement or a call to **getline** with
7365 the same string-valued *expression*. The limit on the number of open *expression*
7366 arguments is implementation-dependent. If the close was successful, the function
7367 shall return zero; otherwise, it shall return non-zero.

7368 *expression* / **getline** [*var*]
7369 Read a record of input from a stream piped from the output of a command. The
7370 stream shall be created if no stream is currently open with the value of *expression* as
7371 its command name. The stream created shall be equivalent to one created by a call
7372 to the *popen*() function with the value of *expression* as the *command* argument and a
7373 value of *r* as the *mode* argument. As long as the stream remains open, subsequent
7374 calls in which *expression* evaluates to the same string value shall read subsequent
7375 records from the stream. The stream shall remain open until the **close** function is
7376 called with an expression that evaluates to the same string value. At that time, the
7377 stream shall be closed as if by a call to the *pclose*() function. If *var* is missing, \$0 and
7378 **NF** shall be set; otherwise, *var* shall be set and, if appropriate, it shall be considered
7379 a numeric string (see **Expressions in awk** on page 190).

7380 The **getline** operator can form ambiguous constructs when there are
7381 unparenthesized operators (including concatenate) to the left of the ' | ' (to the
7382 beginning of the expression containing **getline**). In the context of the '\$'
7383 operator, ' | ' shall behave as if it had a lower precedence than '\$'. The result of
7384 evaluating other operators is unspecified, and portable applications shall
7385 parenthesize properly all such usages.

7386 **getline** Set \$0 to the next input record from the current input file. This form of **getline** shall
7387 set the **NF**, **NR**, and **FNR** variables.

7388 **getline var** Set variable *var* to the next input record from the current input file and, if
7389 appropriate, *var* shall be considered a numeric string (see **Expressions in awk** on
7390 page 190). This form of **getline** shall set the **FNR** and **NR** variables.

getline [*var*] < *expression*

Read the next record of input from a named file. The *expression* shall be evaluated to produce a string that is used as a path name. If the file of that name is not currently open, it shall be opened. As long as the stream remains open, subsequent calls in which *expression* evaluates to the same string value shall read subsequent records from the file. The file shall remain open until the **close** function is called with an expression that evaluates to the same string value. If *var* is missing, \$0 and NF shall be set; otherwise, *var* shall be set and, if appropriate, it shall be considered a numeric string (see **Expressions in awk** on page 190).

The **getline** operator can form ambiguous constructs when there are unparenthesized binary operators (including concatenate) to the right of the '<' (up to the end of the expression containing the **getline**). The result of evaluating such a construct is unspecified, and portable applications shall parenthesize properly all such usages.

system(*expression*)

Execute the command given by *expression* in a manner equivalent to the *system()* function defined in the System Interfaces volume of IEEE Std. 1003.1-200x and return the exit status of the command.

All forms of **getline** shall return 1 for successful input, zero for end-of-file, and -1 for an error.

Where strings are used as the name of a file or pipeline, the application shall ensure that the strings are textually identical. The terminology “same string value” implies that “equivalent strings”, even those that differ only by <space> characters, represent different files.

User-Defined Functions

The *awk* language also provides user-defined functions. Such functions can be defined as:

```
function name([parameter, ...]) { statements }
```

A function can be referred to anywhere in an *awk* program; in particular, its use can precede its definition. The scope of a function is global.

Function parameters, if present, can be either scalars or arrays; the behavior is undefined if an array name is passed as a parameter that the function uses as a scalar, or if a scalar expression is passed as a parameter that the function uses as an array. Function parameters shall be passed by value if scalar and by reference if array name.

The number of parameters in the function definition need not match the number of parameters in the function call. Excess formal parameters can be used as local variables. If fewer arguments are supplied in a function call than are in the function definition, the extra parameters that are used in the function body as scalars shall evaluate to the uninitialized value until they are otherwise initialized, and the extra parameters that are used in the function body as arrays shall be treated as uninitialized arrays where each element evaluates to the uninitialized value until otherwise initialized.

When invoking a function, no white space can be placed between the function name and the opening parenthesis. Function calls can be nested and recursive calls can be made upon functions. Upon return from any nested or recursive function call, the values of all of the calling function's parameters shall be unchanged, except for array parameters passed by reference. The **return** statement can be used to return a value. If a **return** statement appears outside of a function definition, the behavior is undefined.

In the function definition, <newline> characters shall be optional before the opening brace and after the closing brace. Function definitions can appear anywhere in the program where a

7437 *pattern-action* pair is allowed.

7438 Grammar

7439 The grammar in this section and the lexical conventions in the following section shall together
 7440 describe the syntax for *awk* programs. The general conventions for this style of grammar are
 7441 described in Section 1.10 on page 24. A valid program can be represented as the non-terminal
 7442 symbol *program* in the grammar. This formal syntax shall take precedence over the preceding
 7443 text syntax description.

```

7444 %token NAME NUMBER STRING ERE
7445 %token FUNC_NAME      /* Name followed by '(' without white space. */
7446 /* Keywords */
7447 %token      Begin      End
7448 /*          'BEGIN' 'END'                                     */
7449 %token      Break      Continue      Delete      Do      Else
7450 /*          'break' 'continue' 'delete' 'do' 'else' */
7451 %token      Exit      For      Function      If      In
7452 /*          'exit' 'for' 'function' 'if' 'in'      */
7453 %token      Next      Print      Printf      Return      While
7454 /*          'next' 'print' 'printf' 'return' 'while' */
7455 /* Reserved function names */
7456 %token BUILTIN_FUNC_NAME
7457 /* One token for the following:
7458  * atan2 cos sin exp log sqrt int rand srand
7459  * gsub index length match split sprintf sub
7460  * substr tolower toupper close system
7461  */
7462 %token GETLINE
7463 /* Syntactically different from other built-ins. */
7464 /* Two-character tokens. */
7465 %token ADD_ASSIGN SUB_ASSIGN MUL_ASSIGN DIV_ASSIGN MOD_ASSIGN POW_ASSIGN
7466 /*      '+=',      '-=',      '*=',      '/=',      '%=',      '^=' */
7467 %token OR      AND      NO_MATCH      EQ      LE      GE      NE      INCR      DECR      APPEND
7468 /*      '|', '&&', '!~', '==', '<=', '>=', '!=', '++', '--', '>>' */
7469 /* One-character tokens. */
7470 %token '{', '}', '(', ')', '[', ']', ',', ';', NEWLINE
7471 %token '+', '-', '*', '%', '^', '!', '>', '<', '|', '?', ':', '~', '$', '='
7472 %start program
7473 %%
7474 program      : item_list
7475              | actionless_item_list
7476              ;
7477 item_list    : newline_opt
7478              | actionless_item_list item terminator
7479              | item_list item terminator
7480              | item_list action terminator
7481              ;

```



```

7482     actionless_item_list : item_list           pattern terminator
7483                           | actionless_item_list pattern terminator
7484                           ;

7485     item                   : pattern action
7486                           | Function NAME      '(' param_list_opt ')'
7487                             newline_opt action
7488                           | Function FUNC_NAME '(' param_list_opt ')'
7489                             newline_opt action
7490                           ;

7491     param_list_opt        : /* empty */
7492                           | param_list
7493                           ;

7494     param_list            : NAME
7495                           | param_list ',' NAME
7496                           ;

7497     pattern               : Begin
7498                           | End
7499                           | expr
7500                           | expr ',' newline_opt expr
7501                           ;

7502     action                : '{' newline_opt
7503                           | '{' newline_opt terminated_statement_list
7504                           | '{' newline_opt unterminated_statement_list
7505                           ;

7506     terminator            : terminator ';'
7507                           | terminator NEWLINE
7508                           | ';'
7509                           | NEWLINE
7510                           ;

7511     terminated_statement_list : terminated_statement
7512                               | terminated_statement_list terminated_statement
7513                               ;

7514     unterminated_statement_list : unterminated_statement
7515                                   | terminated_statement_list unterminated_statement
7516                                   ;

7517     terminated_statement : action newline_opt
7518                           | If '(' expr ')' newline_opt terminated_statement
7519                           | If '(' expr ')' newline_opt terminated_statement
7520                             Else newline_opt terminated_statement
7521                           | While '(' expr ')' newline_opt terminated_statement
7522                           | For '(' simple_statement_opt ';'
7523                             expr_opt ';' simple_statement_opt ')' newline_opt
7524                             terminated_statement
7525                           | For '(' NAME In NAME ')' newline_opt
7526                             terminated_statement
7527                           | ';' newline_opt
7528                           | terminatable_statement NEWLINE newline_opt
7529                           | terminatable_statement ';'      newline_opt

```

```

7530             ;
7531     unterminated_statement : terminatable_statement
7532         | If '(' expr ')' newline_opt unterminated_statement
7533         | If '(' expr ')' newline_opt terminated_statement
7534         | Else newline_opt unterminated_statement
7535         | While '(' expr ')' newline_opt unterminated_statement
7536         | For '(' simple_statement_opt ';'
7537           expr_opt ';' simple_statement_opt ')' newline_opt
7538           unterminated_statement
7539         | For '(' NAME In NAME ')' newline_opt
7540           unterminated_statement
7541             ;
7542     terminatable_statement : simple_statement
7543         | Break
7544         | Continue
7545         | Next
7546         | Exit expr_opt
7547         | Return expr_opt
7548         | Do newline_opt terminated_statement While '(' expr ')'
7549             ;
7550     simple_statement_opt : /* empty */
7551         | simple_statement
7552             ;
7553     simple_statement : Delete NAME '[' expr_list ']'
7554         | expr
7555         | print_statement
7556             ;
7557     print_statement : simple_print_statement
7558         | simple_print_statement output_redirection
7559             ;
7560     simple_print_statement : Print print_expr_list_opt
7561         | Print '(' multiple_expr_list ')'
7562         | Printf print_expr_list
7563         | Printf '(' multiple_expr_list ')'
7564             ;
7565     output_redirection : '>' expr
7566         | APPEND expr
7567         | '|' expr
7568             ;
7569     expr_list_opt : /* empty */
7570         | expr_list
7571             ;
7572     expr_list : expr
7573         | multiple_expr_list
7574             ;
7575     multiple_expr_list : expr ',' newline_opt expr
7576         | multiple_expr_list ',' newline_opt expr

```

```

7577                                     ;
7578     expr_opt                        : /* empty */
7579                                     | expr
7580                                     ;
7581     expr                            : unary_expr
7582                                     | non_unary_expr
7583                                     ;
7584     unary_expr                      : '+' expr
7585                                     | '-' expr
7586                                     | unary_expr '^'      expr
7587                                     | unary_expr '*'      expr
7588                                     | unary_expr '/'      expr
7589                                     | unary_expr '%'      expr
7590                                     | unary_expr '+'      expr
7591                                     | unary_expr '-'      expr
7592                                     | unary_expr          non_unary_expr
7593                                     | unary_expr '<'      expr
7594                                     | unary_expr LE      expr
7595                                     | unary_expr NE      expr
7596                                     | unary_expr EQ      expr
7597                                     | unary_expr '>'      expr
7598                                     | unary_expr GE      expr
7599                                     | unary_expr '~'      expr
7600                                     | unary_expr NO_MATCH expr
7601                                     | unary_expr In NAME
7602                                     | unary_expr AND newline_opt expr
7603                                     | unary_expr OR  newline_opt expr
7604                                     | unary_expr '?' expr ':' expr
7605                                     | unary_input_function
7606                                     ;
7607     non_unary_expr                  : '(' expr ')'
7608                                     | '!' expr
7609                                     | non_unary_expr '^'      expr
7610                                     | non_unary_expr '*'      expr
7611                                     | non_unary_expr '/'      expr
7612                                     | non_unary_expr '%'      expr
7613                                     | non_unary_expr '+'      expr
7614                                     | non_unary_expr '-'      expr
7615                                     | non_unary_expr          non_unary_expr
7616                                     | non_unary_expr '<'      expr
7617                                     | non_unary_expr LE      expr
7618                                     | non_unary_expr NE      expr
7619                                     | non_unary_expr EQ      expr
7620                                     | non_unary_expr '>'      expr
7621                                     | non_unary_expr GE      expr
7622                                     | non_unary_expr '~'      expr
7623                                     | non_unary_expr NO_MATCH expr
7624                                     | non_unary_expr In NAME
7625                                     | '(' multiple_expr_list ')' In NAME
7626                                     | non_unary_expr AND newline_opt expr

```

```

7627         | non_unary_expr OR newline_opt expr
7628         | non_unary_expr '?' expr ':' expr
7629         | NUMBER
7630         | STRING
7631         | lvalue
7632         | ERE
7633         | lvalue INCR
7634         | lvalue DECR
7635         | INCR lvalue
7636         | DECR lvalue
7637         | lvalue POW_ASSIGN expr
7638         | lvalue MOD_ASSIGN expr
7639         | lvalue MUL_ASSIGN expr
7640         | lvalue DIV_ASSIGN expr
7641         | lvalue ADD_ASSIGN expr
7642         | lvalue SUB_ASSIGN expr
7643         | lvalue '=' expr
7644         | FUNC_NAME '(' expr_list_opt ')'
7645         | /* no white space allowed before '(' */
7646         | BUILTIN_FUNC_NAME '(' expr_list_opt ')'
7647         | BUILTIN_FUNC_NAME
7648         | non_unary_input_function
7649         ;

7650     print_expr_list_opt : /* empty */
7651         | print_expr_list
7652         ;

7653     print_expr_list : print_expr
7654         | print_expr_list ',' newline_opt print_expr
7655         ;

7656     print_expr : unary_print_expr
7657         | non_unary_print_expr
7658         ;

7659     unary_print_expr : '+' print_expr
7660         | '-' print_expr
7661         | unary_print_expr '^' print_expr
7662         | unary_print_expr '*' print_expr
7663         | unary_print_expr '/' print_expr
7664         | unary_print_expr '%' print_expr
7665         | unary_print_expr '+' print_expr
7666         | unary_print_expr '-' print_expr
7667         | unary_print_expr non_unary_print_expr
7668         | unary_print_expr '~' print_expr
7669         | unary_print_expr NO_MATCH print_expr
7670         | unary_print_expr In NAME
7671         | unary_print_expr AND newline_opt print_expr
7672         | unary_print_expr OR newline_opt print_expr
7673         | unary_print_expr '?' print_expr ':' print_expr
7674         ;

7675     non_unary_print_expr : '(' expr ')'
7676         | '!' print_expr

```

```

7677         | non_unary_print_expr '^'      print_expr
7678         | non_unary_print_expr '*'      print_expr
7679         | non_unary_print_expr '/'      print_expr
7680         | non_unary_print_expr '%'      print_expr
7681         | non_unary_print_expr '+'      print_expr
7682         | non_unary_print_expr '-'      print_expr
7683         | non_unary_print_expr         non_unary_print_expr
7684         | non_unary_print_expr '~'      print_expr
7685         | non_unary_print_expr NO_MATCH print_expr
7686         | non_unary_print_expr In NAME
7687         | '(' multiple_expr_list ')' In NAME
7688         | non_unary_print_expr AND newline_opt print_expr
7689         | non_unary_print_expr OR  newline_opt print_expr
7690         | non_unary_print_expr '?' print_expr ':' print_expr
7691         | NUMBER
7692         | STRING
7693         | lvalue
7694         | ERE
7695         | lvalue INCR
7696         | lvalue DECR
7697         | INCR lvalue
7698         | DECR lvalue
7699         | lvalue POW_ASSIGN print_expr
7700         | lvalue MOD_ASSIGN print_expr
7701         | lvalue MUL_ASSIGN print_expr
7702         | lvalue DIV_ASSIGN print_expr
7703         | lvalue ADD_ASSIGN print_expr
7704         | lvalue SUB_ASSIGN print_expr
7705         | lvalue '=' print_expr
7706         | FUNC_NAME '(' expr_list_opt ')'
7707         | /* no white space allowed before '(' */
7708         | BUILTIN_FUNC_NAME '(' expr_list_opt ')'
7709         | BUILTIN_FUNC_NAME
7710         ;
7711
7711     lvalue      : NAME
7712                 | NAME '[' expr_list ']'
7713                 | '$' expr
7714                 ;
7715
7715     non_unary_input_function : simple_get
7716                             | simple_get '<' expr
7717                             | non_unary_expr '|' simple_get
7718                             ;
7719
7719     unary_input_function : unary_expr '|' simple_get
7720                         ;
7721
7721     simple_get      : GETLINE
7722                     | GETLINE lvalue
7723                     ;
7724
7724     newline_opt     : /* empty */
7725                     | newline_opt NEWLINE
7726                     ;

```

This grammar has several ambiguities that shall be resolved as follows:

- Operator precedence and associativity shall be as described in Table 4-1 on page 190.
- In case of ambiguity, an **else** shall be associated with the most immediately preceding **if** that would satisfy the grammar.
- In some contexts, a slash ('/') that is used to surround an ERE could also be the division operator. This shall be resolved in such a way that wherever the division operator could appear, a slash is assumed to be the division operator. (There is no unary division operator.)

One convention that might not be obvious from the formal grammar is where <newline> characters are acceptable. There are several obvious placements such as terminating a statement, and a backslash can be used to escape <newline> characters between any lexical tokens. In addition, <newline> characters without backslashes can follow a comma, an open brace, logical AND operator ("&&"), logical OR operator (" || "), the **do** keyword, the **else** keyword, and the closing parenthesis of an **if**, **for**, or **while** statement. For example:

```
{ print  $1,
        $2 }
```

Lexical Conventions

The lexical conventions for *awk* programs, with respect to the preceding grammar, shall be as follows:

1. Except as noted, *awk* shall recognize the longest possible token or delimiter beginning at a given point.
2. A comment shall consist of any characters beginning with the number sign character and terminated by, but excluding the next occurrence of, a <newline> character. Comments shall have no effect, except to delimit lexical tokens.
3. The <newline> character shall be recognized as the token **NEWLINE**.
4. A backslash character immediately followed by a <newline> character shall have no effect.
5. The token **STRING** shall represent a string constant. A string constant shall begin with the character ' " '. Within a string constant, a backslash character shall be considered to begin an escape sequence as specified in the table in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 5, File Format Notation ('\\' , '\a' , '\b' , '\f' , '\n' , '\r' , '\t' , '\v'). In addition, the escape sequences in Table 4-2 on page 195 shall be recognized. A <newline> character shall not occur within a string constant. A string constant shall be terminated by the first unescaped occurrence of the character ' " ' after the one that begins the string constant. The value of the string shall be the sequence of all unescaped characters and values of escape sequences between, but not including, the two delimiting ' " ' characters.
6. The token **ERE** represents an extended regular expression constant. An ERE constant shall begin with the slash character. Within an ERE constant, a backslash character shall be considered to begin an escape sequence as specified in the table in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 5, File Format Notation. In addition, the escape sequences in Table 4-2 on page 195 shall be recognized. The application shall ensure that a <newline> character does not occur within an ERE constant. An ERE constant shall be terminated by the first unescaped occurrence of the slash character after the one that begins the ERE constant. The extended regular expression represented by the ERE constant shall be the sequence of all unescaped characters and values of escape sequences between, but not including, the two delimiting slash characters.

7. A <blank> character shall have no effect, except to delimit lexical tokens or within **STRING** or **ERE** tokens.

8. The token **NUMBER** shall represent a numeric constant. Its form and numeric value shall be equivalent to either of the tokens **floating-constant** or **integer-constant** as specified by the ISO C standard, with the following exceptions:

- An integer constant cannot begin with 0x or include the hexadecimal digits 'a', 'b', 'c', 'd', 'e', 'f', 'A', 'B', 'C', 'D', 'E', or 'F'.
- The value of an integer constant beginning with 0 shall be taken in decimal rather than octal.
- An integer constant cannot include a suffix ('u', 'U', 'l', or 'L').
- A floating constant cannot include a suffix ('f', 'F', 'l', or 'L').

If the value is too large or too small to be representable, the behavior is undefined.

9. A sequence of underscores, digits, and alphabets from the portable character set (see the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 6.1, Portable Character Set), beginning with an underscore or alphabetic, shall be considered a word.
10. The following words are keywords that shall be recognized as individual tokens; the name of the token is the same as the keyword:

BEGIN	delete	END	function	in	printf
break	do	exit	getline	next	return
continue	else	for	if	print	while

11. The following words are names of built-in functions and shall be recognized as the token **BUILTIN_FUNC_NAME**:

atan2	gsub	log	split	sub	toupper
close	index	match	sprintf	substr	
cos	int	rand	sqrt	system	
exp	length	sin	srand	tolower	

The above-listed keywords and names of built-in functions are considered reserved words.

12. The token **NAME** shall consist of a word that is not a keyword or a name of a built-in function and is not followed immediately (without any delimiters) by the ' (' character.
13. The token **FUNC_NAME** shall consist of a word that is not a keyword or a name of a built-in function, followed immediately (without any delimiters) by the ' (' character. The ' (' character shall not be included as part of the token.
14. The following two-character sequences shall be recognized as the named tokens:

Token Name	Sequence	Token Name	Sequence
ADD_ASSIGN	+=	NO_MATCH	!~
SUB_ASSIGN	-=	EQ	==
MUL_ASSIGN	*=	LE	<=
DIV_ASSIGN	/=	GE	>=
MOD_ASSIGN	%=	NE	!=
POW_ASSIGN	^=	INCR	++
OR	 	DECR	—
AND	&&	APPEND	>>

7814 15. The following single characters shall be recognized as tokens whose names are the
7815 character:

7816 <newline> { } () [] , ; + - * % ^ ! > < | ? : ~ \$ =

7817 There is a lexical ambiguity between the token **ERE** and the tokens **'/'** and **DIV_ASSIGN**.
7818 When an input sequence begins with a slash character in any syntactic context where the token
7819 **'/'** or **DIV_ASSIGN** could appear as the next token in a valid program, the longer of those two
7820 tokens that can be recognized shall be recognized. In any other syntactic context where the token
7821 **ERE** could appear as the next token in a valid program, the token **ERE** shall be recognized.

7822 EXIT STATUS

7823 The following exit values shall be returned:

7824 0 All input files were processed successfully.

7825 >0 An error occurred.

7826 The exit status can be altered within the program by using an **exit** expression.

7827 CONSEQUENCES OF ERRORS

7828 If any *file* operand is specified and the named file cannot be accessed, *awk* shall write a
7829 diagnostic message to standard error and terminate without any further action.

7830 If the program specified by either the *program* operand or a *progfile* operand is not a valid *awk*
7831 program (as specified in the EXTENDED DESCRIPTION section), the behavior is undefined.

7832 APPLICATION USAGE

7833 The **index**, **length**, **match**, and **substr** functions should not be confused with similar functions in
7834 the ISO C standard; the *awk* versions deal with characters, while the ISO C standard deals with
7835 bytes.

7836 Because the concatenation operation is represented by adjacent expressions rather than an
7837 explicit operator, it is often necessary to use parentheses to enforce the proper evaluation
7838 precedence.

7839 EXAMPLES

7840 The *awk* program specified in the command line is most easily specified within single-quotes (for
7841 example, *'program'*) for applications using *sh*, because *awk* programs commonly contain
7842 characters that are special to the shell, including double-quotes. In the cases where an *awk*
7843 program contains single-quote characters, it is usually easiest to specify most of the program as
7844 strings within single-quotes concatenated by the shell with quoted single-quote characters. For
7845 example:

7846 `awk '/\'/ { print "quote:", $0 }'`

7847 prints all lines from the standard input containing a single-quote character, prefixed with *quote*..

7848 The following are examples of simple *awk* programs:

7849 1. Write to the standard output all input lines for which field 3 is greater than 5:

7850 `$3 > 5`

7851 2. Write every tenth line:

7852 `(NR % 10) == 0`

7853 3. Write any line with a substring matching the regular expression:

7854 `/(G|D)(2[0-9][[:alpha:]]*)/`


```

7855 4. Print any line with a substring containing a 'G' or 'D', followed by a sequence of digits
7856 and characters. This example uses character classes digit and alpha to match language-
7857 independent digit and alphabetic characters respectively:
7858 /(G|D)([[:digit:][:alpha:]]*)/
7859
7860 5. Write any line in which the second field matches the regular expression and the fourth
7861 field does not:
7862 $2 ~ /xyz/ && $4 !~ /xyz/
7863
7864 6. Write any line in which the second field contains a backslash:
7865 $2 ~ /\\"/>

```

7894 }
 7895 18. Write the path prefixes contained in the *PATH* environment variable, one per line:

```
7896       BEGIN {
7897            n = split (ENVIRON["PATH"], path, ":")
7898            for (i = 1; i <= n; ++i)
7899               print path[i]
7900       }
```

7901 19. If there is a file named **input** containing page headers of the form:

7902 Page #
 7903 and a file named **program** that contains:

```
7904       /Page/   { $2 = n++; }
7905               { print }
```

7906 then the command line:

```
7907       awk -f program n=5 input
```

7908 prints the file **input**, filling in page numbers starting at 5.

7909 RATIONALE

7910 In **sub** and **gsub**, if *repl* is a string literal (the lexical token **STRING**), then two consecutive
 7911 backslash characters should be used in the string to ensure a single backslash will precede the
 7912 ampersand when the resultant string is passed to the function. (For example, to specify one
 7913 literal ampersand in the replacement string, use **gsub(ERE, "\\&")**.)

7914 Historically the only special character in the *repl* argument of **sub** and **gsub** string functions was
 7915 the ampersand ('&') character and preceding it with the backslash character was used to turn
 7916 off its special meaning.

7917 The description in the ISO POSIX-2:1993 standard introduced behavior such that the backslash
 7918 character was another special character and it was unspecified whether there were any other
 7919 special characters. This description introduced several portability problems, some of which are
 7920 described below, and so it has been replaced with the more historical description. Some of the
 7921 problems include:

- 7922 • Historically, to create the replacement string, a script could use **gsub(ERE, "\\&")**, but with
 7923 the ISO POSIX-2:1993 standard wording, it was necessary to use **gsub(ERE, "\\&")**.
 7924 Backslash characters are doubled here because all string literals are subject to lexical analysis,
 7925 which would reduce each pair of backslash characters to a single backslash before being
 7926 passed to **gsub**.
- 7927 • Since it was unspecified what the special characters were, for portable scripts to guarantee
 7928 that characters are printed literally, each character had to be preceded with a backslash. (For
 7929 example, a portable script had to use **gsub(ERE, "\\h\\i")** to produce a replacement string
 7930 of "hi".)

7931 The description for comparisons in the ISO POSIX-2:1993 standard did not properly describe
 7932 historical practice because of the way numeric strings are compared as numbers. The current
 7933 rules cause the following code:

```
7934       if (0 == "000")
7935           print "strange, but true"
7936       else
7937           print "not true"
```

to do a numeric comparison, causing the **if** to succeed. It should be intuitively obvious that this is incorrect behavior, and indeed, no historical implementation of *awk* actually behaves this way.

To fix this problem, the definition of *numeric string* was enhanced to include only those values obtained from specific circumstances (mostly external sources) where it is not possible to determine unambiguously whether the value is intended to be a string or a numeric.

Variables that are assigned to a numeric string shall also be treated as a numeric string. (For example, the notion of a numeric string can be propagated across assignments.) In comparisons, all variables having the uninitialized value are to be treated as a numeric operand evaluating to the numeric value zero.

Uninitialized variables include all types of variables including scalars, array elements, and fields. The definition of an uninitialized value is necessary to describe the value placed on uninitialized variables and on fields that are valid (for example, **<\$NF**) but have no characters in them and to describe how these variables are to be used in comparisons. A valid field, such as **\$1**, that has no characters in it can be obtained by from an input line of `"\t\t"` when **FS** = `"\t"`. Historically, the comparison (**\$1**<**10**) was done numerically after evaluating **\$1** to the value zero.

The phrase "... also shall have the numeric value of the numeric string" was removed from several sections of the ISO POSIX-2 standard because they specify an unnecessary implementation detail. It is not necessary for IEEE Std. 1003.1-200x to specify that these objects be assigned two different values. It is only necessary to specify that these objects may evaluate to two different values depending on context.

The description of numeric string processing is based on the behavior of the *atof()* function in the ISO C standard. While it is not a requirement for an implementation to use this function, many historical implementations of *awk* do. In the ISO C standard, floating point constants use a period as a decimal point character for the language itself, independent of the current locale, but the *atof()* function and the associated *strtod()* function use the decimal point character of the current locale when converting strings to numeric values. Similarly in *awk*, floating point constants in an *awk* script use a period independent of the locale, but input strings use the decimal point character of the locale.

The ISO POSIX-2 standard description is based on the new *awk*, "nawk", (see the referenced *The AWK Programming Language*), which introduced a number of new features to the historical *awk*:

1. New keywords: **delete**, **do**, **functin**, **return**
2. New built-in functions: **atan2**, **close**, **cos**, **gsub**, **match**, **rand**, **sin**, **srand**, **sub**, **system**
3. New predefined variables: **FNR**, **ARGC**, **ARGV**, **RSTART**, **RLENGTH**, **SUBSEP**
4. New expression operators: **?**, **:**, **..**, **^**
5. The **FS** variable and the third argument to **split**, now treated as extended regular expressions.
6. The operator precedence, changed to more closely match the C language. Two examples of code that operate differently are:

```
while ( n /= 10 > 1 ) ...
if (!"wk" ~ /bwk/) ...
```

Several features have been added based on newer implementations of *awk*:

- Multiple instances of **-f progfile** are permitted
- The new option **-v assignment**

- The new predefined variable **ENVIRON**
- New built-in functions **toupper**, and **tolower**
- More formatting capabilities are added to **printf** to match the ISO C standard

The overall *awk* syntax has always been based on the C language, with a few features from the shell command language and other sources. Because of this, it is not completely compatible with any other language, which has caused confusion for some users. It is not the intent of the standard developers to address such issues. IEEE Std. 1003.1-200x has made a few relatively minor changes toward making the language more compatible with the C language as specified by the ISO C standard; most of these changes are based on similar changes in recent implementations, as described above. There remain several C-language conventions that are not in *awk*. One of the notable ones is the comma operator, which is commonly used to specify multiple expressions in the C language **for** statement. Also, there are various places where *awk* is more restrictive than the C language regarding the type of expression that can be used in a given context. These limitations are due to the different features that the *awk* language does provide.

Regular expressions in *awk* have been extended somewhat from historical implementations to make them a pure superset of extended regular expressions, as defined by IEEE Std. 1003.1-200x (see the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 9.4, Extended Regular Expressions). The main extensions are internationalization features and interval expressions. Historical implementations of *awk* have long supported backslash escape sequences as an extension to extended regular expressions, and this extension has been retained despite inconsistency with other utilities. The number of escape sequences recognized in both extended regular expressions and strings has varied (generally increasing with time) among implementations. The set specified by IEEE Std. 1003.1-200x includes most sequences known to be supported by popular implementations and by the ISO C standard. One sequence that is not supported is hexadecimal value escapes beginning with '`\x`'. This would allow values expressed in more than 9 bits to be used within *awk* as in the ISO C standard. However, because this syntax has a non-deterministic length, it does not permit the subsequent character to be a hexadecimal digit. This limitation can be dealt with in the C language by the use of lexical string concatenation. In the *awk* language, concatenation could also be a solution for strings, but not for extended regular expressions (either lexical ERE tokens or strings used dynamically as regular expressions). Because of this limitation, the feature has not been added to IEEE Std. 1003.1-200x.

When a string variable is used in a context where an extended regular expression normally appears (where the lexical token ERE is used in the grammar) the string does not contain the literal slashes.

Some versions of *awk* allow the form:

```
func name(args, ... ) { statements }
```

This has been deprecated by the authors of the language, who asked that it not be included in IEEE Std. 1003.1-200x.

Historical implementations of *awk* produce an error if a **next** statement is executed in a **BEGIN** action, and cause *awk* to terminate if a **next** statement is executed in an **END** action. This behavior has not been documented, and it was not believed that it was necessary to standardize it.

The specification of conversions between string and numeric values is much more detailed than in the documentation of historical implementations or in the referenced *The AWK Programming Language*. Although most of the behavior is designed to be intuitive, the details are necessary to ensure compatible behavior from different implementations. This is especially important in relational expressions since the types of the operands determine whether a string or numeric

comparison is performed. From the perspective of an application writer, it is usually sufficient to expect intuitive behavior and to force conversions (by adding zero or concatenating a null string) when the type of an expression does not obviously match what is needed. The intent has been to specify historical practice in almost all cases. The one exception is that, in historical implementations, variables and constants maintain both string and numeric values after their original value is converted by any use. This means that referencing a variable or constant can have unexpected side effects. For example, with historical implementations the following program:

```

{
    a = "+2"
    b = 2
    if (NR % 2)
        c = a + b
    if (a == b)
        print "numeric comparison"
    else
        print "string comparison"
}
```

would perform a numeric comparison (and output numeric comparison) for each odd-numbered line, but perform a string comparison (and output string comparison) for each even-numbered line. IEEE Std. 1003.1-200x ensures that comparisons will be numeric if necessary. With historical implementations, the following program:

```

BEGIN {
    OFMT = "%e"
    print 3.14
    OFMT = "%f"
    print 3.14
}
```

would output "3.140000e+00" twice, because in the second **print** statement the constant "3.14" would have a string value from the previous conversion. IEEE Std. 1003.1-200x requires that the output of the second **print** statement be "3.140000". The behavior of historical implementations was seen as too unintuitive and unpredictable.

It was pointed out that with the rules contained in early drafts, the following script would print nothing:

```

BEGIN {
    y[1.5] = 1
    OFMT = "%e"
    print y[1.5]
}
```

Therefore, a new variable, **CONVFMT**, was introduced. The **OFMT** variable is now restricted to affecting output conversions of numbers to strings and **CONVFMT** is used for internal conversions, such as comparisons or array indexing. The default value is the same as that for **OFMT**, so unless a program changes **CONVFMT** (which no historical program would do), it will receive the historical behavior associated with internal string conversions.

The POSIX *awk* lexical and syntactic conventions are specified more formally than in other sources. Again the intent has been to specify historical practice. One convention that may not be obvious from the formal grammar as in other verbal descriptions is where <newline> characters are acceptable. There are several obvious placements such as terminating a statement, and a

backslash can be used to escape <newline> characters between any lexical tokens. In addition, <newline> characters without backslashes can follow a comma, an open brace, a logical AND operator ("&&"), a logical OR operator ("||"), the **do** keyword, the **else** keyword, and the closing parenthesis of an **if**, **for**, or **while** statement. For example:

```
{ print $1,
    $2 }
```

The requirement that *awk* add a trailing <newline> character to the program argument text is to simplify the grammar, making it match a text file in form. There is no way for an application or test suite to determine whether a literal <newline> is added or whether *awk* simply acts as if it did.

IEEE Std. 1003.1-200x requires several changes from historical implementations in order to support internationalization. Probably the most subtle of these is the use of the decimal-point character, defined by the *LC_NUMERIC* category of the locale, in representations of floating-point numbers. This locale-specific character is used in recognizing numeric input, in converting between strings and numeric values, and in formatting output. However, regardless of locale, the period character (the decimal-point character of the POSIX locale) is the decimal-point character recognized in processing *awk* programs (including assignments in command line arguments). This is essentially the same convention as the one used in the ISO C standard. The difference is that the C language includes the *setlocale()* function, which permits an application to modify its locale. Because of this capability, a C application begins executing with its locale set to the C locale, and only executes in the environment-specified locale after an explicit call to *setlocale()*. However, adding such an elaborate new feature to the *awk* language was seen as inappropriate for IEEE Std. 1003.1-200x. It is possible to execute an *awk* program explicitly in any desired locale by setting the environment in the shell.

The undefined behavior resulting from NULs in extended regular expressions allows future extensions for the GNU *gawk* program to process binary data.

The behavior in the case of invalid *awk* programs (including lexical, syntactic, and semantic errors) is undefined because it was considered overly limiting on implementations to specify. In most cases such errors can be expected to produce a diagnostic and a non-zero exit status. However, some implementations may choose to extend the language in ways that make use of certain invalid constructs. Other invalid constructs might be deemed worthy of a warning, but otherwise cause some reasonable behavior. Still other constructs may be very difficult to detect in some implementations. Also, different implementations might detect a given error during an initial parsing of the program (before reading any input files) while others might detect it when executing the program after reading some input. Implementors should be aware that diagnosing errors as early as possible and producing useful diagnostics can ease debugging of applications, and thus make an implementation more usable.

The unspecified behavior from using multi-character **RS** values is to allow possible future extensions based on extended regular expressions used for record separators. Historical implementations take the first character of the string and ignore the others.

Unspecified behavior when *split(string,array,<null>)* is used is to allow a proposed future extension that would split up a string into an array of individual characters.

Because the concatenation operation is represented by adjacent expressions rather than an explicit operator, it is often necessary to use parentheses to enforce the proper evaluation precedence.

In the context of the **getline** function, equally good arguments for different precedences of the **and** and **<** operators can be made. Historical practice has been that:

8123 `getline < "a" "b"`

8124 is parsed as:

8125 `(getline < "a") "b"`

8126 although many would argue that the intent was that the file **ab** should be read. However:

8127 `getline < "x" + 1`

8128 parses as:

8129 `getline < ("x" + 1)`

8130 Similar problems occur with the `|` version of **getline**, particularly in combination with **\$**. For
8131 example:

8132 `$"echo hi" | getline`

8133 (This situation is particularly problematic when used in a **print** statement, where the `|getline`
8134 part might be a redirection of the **print**.)

8135 Since in most cases such constructs are not (or at least should not) be used (because they have a
8136 natural ambiguity for which there is no conventional parsing), the meaning of these constructs
8137 has been made explicitly unspecified. (The effect is that a portable application that runs into the
8138 problem must parenthesize to resolve the ambiguity.) There appeared to be few if any actual
8139 uses of such constructs.

8140 Grammars can be written that would cause an error under these circumstances. Where
8141 backwards compatibility is not a large consideration, implementors may wish to use such
8142 grammars.

8143 Some historical implementations have allowed some built-in functions to be called without an
8144 argument list, the result being a default argument list chosen in some “reasonable” way. Use of
8145 **length** as a synonym for **length(\$0)** is the only one of these forms that is thought to be widely
8146 known or widely used; this particular form is documented in various places (for example, most
8147 historical *awk* reference pages, although not in the referenced *The AWK Programming Language*)
8148 as legitimate practice. With this exception, default argument lists have always been
8149 undocumented and vaguely defined, and it is not at all clear how (or if) they should be
8150 generalized to user-defined functions. They add no useful functionality and preclude possible
8151 future extensions that might need to name functions without calling them. Not standardizing
8152 them seems the simplest course. The standard developers considered that **length** merited special
8153 treatment, however, since it has been documented in the past and sees possibly substantial use
8154 in historical programs. Accordingly, this usage has been made legitimate, for backwards
8155 compatibility, but marked obsolescent in hopes that it can eventually be removed from the
8156 language.

8157 In **sub** and **gsub**, if *repl* is a string literal (the lexical token **STRING**), then two consecutive
8158 backslash characters should be used in the string to ensure a single backslash will precede the
8159 ampersand when the resultant string is passed to the function. (For example, to specify one
8160 literal ampersand in the replacement string, use **gsub(ERE, "\\&")**.)

8161 Historically the only special character in the *repl* argument of **sub** and **gsub** string functions was
8162 the ampersand (**&**) character and preceding it with the backslash character was used to turn
8163 off its special meaning.

8164 The description in the ISO POSIX-2:1993 standard introduced behavior such that the backslash
8165 character was another special character and it was unspecified whether there were any other
8166 special characters. This description introduced several portability problems, some of which are
8167 described below, and so it has been replaced with the more historical description. Some of the

problems include:

- Historically, to create the replacement string, a script could use `gsub(ERE, "\\&")`, but with the ISO POSIX-2:1993 standard wording, it was necessary to use `gsub(ERE, "\\&")`. Backslash characters are doubled here because all string literals are subject to lexical analysis, which would reduce each pair of backslash characters to a single backslash before being passed to `gsub`.
- Since it was unspecified what the special characters were, for portable scripts to guarantee that characters are printed literally, each character had to be preceded with a backslash. (For example, a portable script had to use `gsub(ERE, "\\h\\i")` to produce a replacement string of "hi".)

The description for comparisons in the ISO POSIX-2:1993 standard did not properly describe historical practice because of the way numeric strings are compared as numbers. The current rules cause the following code:

```
if (0 == "000")
    print "strange, but true"
else
    print "not true"
```

to do a numeric comparison, causing the `if` to succeed. It should be intuitively obvious that this is incorrect behavior, and indeed, no historical implementation of `awk` actually behaves this way.

To fix this problem, the definition of *numeric string* was enhanced to include only those values obtained from specific circumstances (mostly external sources) where it is not possible to determine unambiguously whether the value is intended to be a string or a numeric.

Variables that are assigned to a numeric string shall also be treated as a numeric string. (For example, the notion of a numeric string can be propagated across assignments.) In comparisons, all variables having the uninitialized value are to be treated as a numeric operand evaluating to the numeric value zero.

Uninitialized variables include all types of variables including scalars, array elements, and fields. The definition of an uninitialized value in **Variables and Special Variables** on page 193 is necessary to describe the value placed on uninitialized variables and on fields that are valid (for example, `< $NF`) but have no characters in them and to describe how these variables are to be used in comparisons. A valid field, such as `$1`, that has no characters in it can be obtained by from an input line of `"\t\t"` when `FS='\t'`. Historically, the comparison (`$1<10`) was done numerically after evaluating `$1` to the value zero.

The phrase "... also shall have the numeric value of the numeric string" was removed from several sections of the ISO POSIX-2:1993 standard because it specifies an unnecessary implementation detail. It is not necessary for IEEE Std. 1003.1-200x to specify that these objects be assigned two different values. It is only necessary to specify that these objects may evaluate to two different values depending on context.

The description of numeric string processing is based on the behavior of the `atof()` function in the ISO C standard. While it is not a requirement for an implementation to use this function, many historical implementations of `awk` do. In the ISO C standard, floating-point constants use a period as a decimal point character for the language itself, independent of the current locale, but the `atof()` function and the associated `strtod()` function use the decimal point character of the current locale when converting strings to numeric values. Similarly in `awk`, floating point constants in an `awk` script use a period independent of the locale, but input strings use the decimal point character of the locale.

8214 **FUTURE DIRECTIONS**

8215 None.

8216 **SEE ALSO**8217 *grep*, *lex*, *sed*, the System Interfaces volume of IEEE Std. 1003.1-200x, *atof()*, *setlocale()*, *strtod()*8218 **CHANGE HISTORY**

8219 First released in Issue 2.

8220 **Issue 4**

8221 Aligned with the ISO/IEC 9945-2: 1993 standard.

8222 **Issue 4, Version 2**

8223 The EXAMPLES section is corrected as follows:

- 8224 • In Example 10, the braces are removed.
- 8225 • In Example 17, the invocation of **printf** is corrected.

8226 **Issue 5**

8227 FUTURE DIRECTIONS section added.

8228 **Issue 6**8229 The *awk* utility is aligned with the IEEE P1003.2b draft standard.

8230 The normative text is reworded to avoid use of the term “must” for application requirements.

8231 NAME

8232 basename — return non-directory portion of a path name

8233 SYNOPSIS

8234 basename *string* [*suffix*]

8235 DESCRIPTION

8236 The *string* operand shall be treated as a path name, as defined in the System Interface Definitions
8237 volume of IEEE Std. 1003.1-200x, Section 3.272, Path Name. The string *string* shall be converted
8238 to the file name corresponding to the last path name component in *string* and then the suffix
8239 string *suffix*, if present, shall be removed. This shall be done by performing actions equivalent to
8240 the following steps in order:

- 8241 1. If *string* is a null string, it is unspecified whether the resulting string is ' . ' or a null string.
8242 In either case, skip steps 2 through 6.
- 8243 2. If *string* is "/", it is implementation-dependent whether steps 3 to 6 are skipped or
8244 processed.
- 8245 3. If *string* consists entirely of slash characters, *string* shall be set to a single slash character. In
8246 this case, skip steps 4 to 6.
- 8247 4. If there are any trailing slash characters in *string*, they shall be removed.
- 8248 5. If there are any slash characters remaining in *string*, the prefix of *string* up to and including
8249 the last slash character in *string* shall be removed.
- 8250 6. If the *suffix* operand is present, is not identical to the characters remaining in *string*, and is
8251 identical to a suffix of the characters remaining in *string*, the suffix *suffix* shall be removed
8252 from *string*. Otherwise, *string* is modified by this step. It shall not be considered an error if
8253 *suffix* is not found in *string*.

8254 The resulting string shall be written to standard output.

8255 OPTIONS

8256 None.

8257 OPERANDS

8258 The following operands shall be supported:

8259 *string* A string.
8260 *suffix* A string.

8261 STDIN

8262 Not used.

8263 INPUT FILES

8264 None.

8265 ENVIRONMENT VARIABLES

8266 The following environment variables shall affect the execution of *basename*:

8267 *LANG* Provide a default value for the internationalization variables that are unset or null.
8268 If *LANG* is unset or null, the corresponding value from the implementation-
8269 dependent default locale shall be used. If any of the internationalization variables
8270 contains an invalid setting, the utility shall behave as if none of the variables had
8271 been defined.

8272 *LC_ALL* If set to a non-empty string value, override the values of all the other
8273 internationalization variables.

8274 **LC_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as
 8275 characters (for example, single-byte as opposed to multi-byte characters in
 8276 arguments).

8277 **LC_MESSAGES**
 8278 Determine the locale that should be used to affect the format and contents of
 8279 diagnostic messages written to standard error.

8280 xsi **NLSPATH** Determine the location of message catalogs for the processing of **LC_MESSAGES**.

8281 **ASYNCHRONOUS EVENTS**
 8282 Default.

8283 **STDOUT**
 8284 The *basename* utility shall write a line to the standard output in the following format:
 8285 "%s\n", <resulting string>

8286 **STDERR**
 8287 Used only for diagnostic messages.

8288 **OUTPUT FILES**
 8289 None.

8290 **EXTENDED DESCRIPTION**
 8291 None.

8292 **EXIT STATUS**
 8293 The following exit values shall be returned:
 8294 0 Successful completion.
 8295 >0 An error occurred.

8296 **CONSEQUENCES OF ERRORS**
 8297 Default.

8298 **APPLICATION USAGE**
 8299 The definition of *pathname* specifies implementation-dependent behavior for path names starting
 8300 with two slash characters. Therefore, applications shall not arbitrarily add slashes to the
 8301 beginning of a path name unless they can ensure that there are more or less than two or are
 8302 prepared to deal with the implementation-dependent consequences.

8303 **EXAMPLES**
 8304 If the string *string* is a valid path name:
 8305 \$(basename "*string*")
 8306 produces a file name that could be used to open the file named by *string* in the directory
 8307 returned by:
 8308 \$(dirname "*string*")
 8309 If the string *string* is not a valid path name, the same algorithm is used, but the result need not be
 8310 a valid file name. The *basename* utility is not expected to make any judgements about the validity
 8311 of *string* as a path name; it just follows the specified algorithm to produce a result string.
 8312 The following shell script compiles **/usr/src/cmd/cat.c** and moves the output to a file named **cat**
 8313 in the current directory when invoked with the argument **/usr/src/cmd/cat** or with the argument
 8314 **/usr/src/cmd/cat.c**:

```
8315      c89 $(dirname "$1")/$(basename "$1" .c).c
8316      mv a.out $(basename "$1" .c)
```

8317 RATIONALE

8318 The behaviors of *basename* and *dirname* have been coordinated so that when *string* is a valid path
8319 name:

```
8320      $(basename "string")
```

8321 would be a valid file name for the file in the directory:

```
8322      $(dirname "string")
```

8323 This would not work for the early proposal versions of these utilities due to the way it specified
8324 handling of trailing slashes.

8325 Since the definition of *pathname* specifies implementation-dependent behavior for path names
8326 starting with two slash characters, this volume of IEEE Std. 1003.1-200x specifies similar
8327 implementation-dependent behavior for the *basename* and *dirname* utilities.

8328 FUTURE DIRECTIONS

8329 None.

8330 SEE ALSO

8331 *dirname*, Section 2.5 on page 43

8332 CHANGE HISTORY

8333 First released in Issue 2.

8334 Issue 4

8335 Aligned with the ISO/IEC 9945-2: 1993 standard.

8336 Issue 6

8337 PASC Interpretation 1003.2-92 #164 has been applied. |

8338 The normative text is reworded to avoid use of the term “must” for application requirements. |

8339 **NAME**

8340 batch — schedule commands to be executed in a batch queue

8341 **SYNOPSIS**8342 UP *batch*

8343

8344 **DESCRIPTION**8345 The *batch* utility shall read commands from standard input and schedule them for execution in a
8346 batch queue. It shall be the equivalent of the command:

8347 at -q b -m now

8348 where queue *b* is a special *at* queue, specifically for batch jobs. Batch jobs shall be submitted to
8349 the batch queue with no time constraints and shall be run by the system using algorithms, based
8350 on unspecified factors, that may vary with each invocation of *batch*.8351 XSI Users are permitted to use *batch* if their name appears in the file */usr/lib/cron/at.allow*. If that file
8352 does not exist, the file */usr/lib/cron/at.deny* is checked to determine whether the user should be
8353 denied access to *batch*. If neither file exists, only a process with the appropriate privileges is
8354 allowed to submit a job. If only *at.deny* exists and is empty, global usage is permitted. The
8355 *at.allow* and *at.deny* files consist of one user name per line.8356 **OPTIONS**

8357 None.

8358 **OPERANDS**

8359 None.

8360 **STDIN**8361 The standard input shall be a text file consisting of commands acceptable to the shell command
8362 language described in Chapter 2 on page 35.8363 **INPUT FILES**8364 XSI The text files */usr/lib/cron/at.allow* and */usr/lib/cron/at.deny* contain user names, one per line, of
8365 users who are, respectively, authorized or denied access to the *at* and *batch* utilities.8366 **ENVIRONMENT VARIABLES**8367 The following environment variables shall affect the execution of *batch*:8368 *LANG* Provide a default value for the internationalization variables that are unset or null.
8369 If *LANG* is unset or null, the corresponding value from the implementation-
8370 dependent default locale shall be used. If any of the internationalization variables
8371 contains an invalid setting, the utility shall behave as if none of the variables had
8372 been defined.8373 *LC_ALL* If set to a non-empty string value, override the values of all the other
8374 internationalization variables.8375 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
8376 characters (for example, single-byte as opposed to multi-byte characters in
8377 arguments and input files).8378 *LC_MESSAGES*8379 Determine the locale that should be used to affect the format and contents of
8380 diagnostic messages written to standard error and informative messages written to
8381 standard output.8382 *LC_TIME* Determine the format and contents for date and time strings written by *batch*.

8383	XSI	NLSPATH	Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .
8384		SHELL	Determine the name of a command interpreter to be used to invoke the at-job. If
8385			the variable is unset or null, <i>sh</i> shall be used. If it is set to a value other than a name
8386			for <i>sh</i> , the implementation shall do one of the following: use that shell; use <i>sh</i> ; use
8387			the login shell from the user database; any of the preceding accompanied by a
8388			warning diagnostic about which was chosen.
8389		TZ	Determine the timezone. The job shall be submitted for execution at the time
8390			specified by <i>timespec</i> or <i>-t time</i> relative to the timezone specified by the <i>TZ</i>
8391			variable. If <i>timespec</i> specifies a timezone, it overrides <i>TZ</i> . If <i>timespec</i> does not
8392			specify a timezone and <i>TZ</i> is unset or null, an unspecified default timezone shall
8393			be used.
8394		ASYNCHRONOUS EVENTS	
8395			Default.
8396		STDOUT	
8397			When standard input is a terminal, prompts of unspecified format for each line of the user input
8398			described in the STDIN section may be written to standard output.
8399		STDERR	
8400			The following shall be written to standard error when a job has been successfully submitted:
8401			"job %s at %s\n", <i>at_job_id</i> , <date>
8402			where <i>date</i> shall be equivalent in format to the output of:
8403			date +"%a %b %e %T %Y"
8404			The date and time written shall be adjusted so that they appear in the timezone of the user (as
8405			determined by the <i>TZ</i> variable).
8406			Neither this, nor warning messages concerning the selection of the command interpreter, are
8407			considered a diagnostic that changes the exit status.
8408			Diagnostic messages, if any, shall be written to standard error.
8409		OUTPUT FILES	
8410			None.
8411		EXTENDED DESCRIPTION	
8412			None.
8413		EXIT STATUS	
8414			The following exit values shall be returned:
8415			0 Successful completion.
8416			>0 An error occurred.
8417		CONSEQUENCES OF ERRORS	
8418			The job shall not be scheduled.

8419 **APPLICATION USAGE**

8420 It may be useful to redirect standard output within the specified commands.

8421 Application writers should note that this utility need not be provided on systems that do not
8422 support the User Portability Utilities option.8423 **EXAMPLES**

8424 1. This sequence can be used at a terminal:

8425 batch
8426 sort < file >outfile
8427 EOT8428 2. This sequence, which demonstrates redirecting standard error to a pipe, is useful in a
8429 command procedure (the sequence of output redirection specifications is significant):

8430 batch <<! diff file1 file2 2>&1 >outfile | mailx mygroup !

8431 **RATIONALE**8432 Early proposals described *batch* in a manner totally separated from *at*, even though the historical
8433 model treated it almost as a synonym for *at -qb*. A number of features were added to list and
8434 control batch work separately from those in *at*. Upon further reflection, it was decided that the
8435 benefit of this did not merit the change to the historical interface.8436 The **-m** option was included on the equivalent *at* command because it is historical practice to
8437 mail results to the submitter, even if all job-produced output is redirected. As explained in the
8438 RATIONALE for *at*, the **now** keyword submits the job for immediate execution (after scheduling
8439 delays), despite some historical systems where *at now* would have been considered an error.8440 **FUTURE DIRECTIONS**

8441 None.

8442 **SEE ALSO**8443 *at*8444 **CHANGE HISTORY**

8445 First released in Issue 2.

8446 **Issue 4**8447 Format reorganized and separated from the *at* description.

8448 Aligned with the ISO/IEC 9945-2:1993 standard.

8449 **Issue 6**

8450 This utility is now marked as part of the User Portability Utilities option. |

8451 The NAME is changed to align with the IEEE P1003.2b draft standard. |

8452 The normative text is reworded to avoid use of the term “must” for application requirements. |

8453 **NAME**

8454 bc — arbitrary-precision arithmetic language

8455 **SYNOPSIS**8456 bc [-l] [*file* ...]8457 **DESCRIPTION**

8458 The *bc* utility shall implement an arbitrary precision calculator. It shall take input from any files
 8459 given, then read from the standard input. If the standard input and standard output to *bc* are
 8460 attached to a terminal, the invocation of *bc* shall be considered to be *interactive*, causing
 8461 behavioral constraints described in the following sections.

8462 **OPTIONS**

8463 The *bc* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,
 8464 Section 12.2, Utility Syntax Guidelines.

8465 The following option shall be supported:

8466 -l (The letter ell.) Define the math functions and initialize *scale* to 20, instead of the
 8467 default zero; see the EXTENDED DESCRIPTION section.

8468 **OPERANDS**

8469 The following operand shall be supported:

8470 *file* A path name of a text file containing *bc* program statements. After all *files* have
 8471 been read, *bc* shall read the standard input.

8472 **STDIN**

8473 See the INPUT FILES section.

8474 **INPUT FILES**

8475 Input files shall be text files containing a sequence of comments, statements, and function
 8476 definitions that shall be executed as they are read.

8477 **ENVIRONMENT VARIABLES**8478 The following environment variables shall affect the execution of *bc*:

8479 *LANG* Provide a default value for the internationalization variables that are unset or null.
 8480 If *LANG* is unset or null, the corresponding value from the implementation-
 8481 dependent default locale shall be used. If any of the internationalization variables
 8482 contains an invalid setting, the utility shall behave as if none of the variables had
 8483 been defined.

8484 *LC_ALL* If set to a non-empty string value, override the values of all the other
 8485 internationalization variables.

8486 *LC_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as
 8487 characters (for example, single-byte as opposed to multi-byte characters in
 8488 arguments and input files).

8489 *LC_MESSAGES*

8490 Determine the locale that should be used to affect the format and contents of
 8491 diagnostic messages written to standard error.

8492 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC_MESSAGES*.

8493 **ASYNCHRONOUS EVENTS**

8494 Default.

8495 **STDOUT**

8496 The output of the *bc* utility shall be controlled by the program read, and consist of zero or more
 8497 lines containing the value of all executed expressions without assignments. The radix and
 8498 precision of the output shall be controlled by the values of the **obase** and **scale** variables; see the
 8499 EXTENDED DESCRIPTION section.

8500 **STDERR**

8501 Used only for diagnostic messages.

8502 **OUTPUT FILES**

8503 None.

8504 **EXTENDED DESCRIPTION**8505 **Grammar**

8506 The grammar in this section and the lexical conventions in the following section shall together
 8507 describe the syntax for *bc* programs. The general conventions for this style of grammar are
 8508 described in Section 1.10 on page 24. A valid program can be represented as the non-terminal
 8509 symbol **program** in the grammar. This formal syntax shall take precedence over the text syntax
 8510 description.

```

8511 %token      EOF NEWLINE STRING LETTER NUMBER
8512 %token      MUL_OP
8513 /*          '*', '/', '%'                               */
8514 %token      ASSIGN_OP
8515 /*          '=', '+=', '-=', '*=', '/=', '%=', '^=' */
8516 %token      REL_OP
8517 /*          '==', '<=', '>=', '!=', '<', '>'           */
8518 %token      INCR_DECR
8519 /*          '++', '--'                                   */
8520 %token      Define      Break      Quit      Length
8521 /*          'define', 'break', 'quit', 'length'         */
8522 %token      Return      For        If        While      Sqrt
8523 /*          'return', 'for', 'if', 'while', 'sqrt'      */
8524 %token      Scale       Ibase      Obase      Auto
8525 /*          'scale', 'ibase', 'obase', 'auto'          */
8526 %start      program
8527 %%
8528 program      : EOF
8529               | input_item program
8530               ;
8531 input_item    : semicolon_list NEWLINE
8532               | function
8533               ;
8534 semicolon_list : /* empty */
8535               | statement
8536               | semicolon_list ';' statement
8537               | semicolon_list ';'

```

```

8538                                     ;
8539     statement_list                   : /* empty */
8540                                     | statement
8541                                     | statement_list NEWLINE
8542                                     | statement_list NEWLINE statement
8543                                     | statement_list ';'
8544                                     | statement_list ';' statement
8545                                     ;
8546     statement                        : expression
8547                                     | STRING
8548                                     | Break
8549                                     | Quit
8550                                     | Return
8551                                     | Return '(' return_expression ')'
8552                                     | For '(' expression ';'
8553                                         relational_expression ';'
8554                                         expression ')' statement
8555                                     | If '(' relational_expression ')' statement
8556                                     | While '(' relational_expression ')' statement
8557                                     | '{' statement_list '}'
8558                                     ;
8559     function                          : Define LETTER '(' opt_parameter_list ')'
8560                                     | '{' NEWLINE opt_auto_define_list
8561                                     | statement_list '}'
8562                                     ;
8563     opt_parameter_list                : /* empty */
8564                                     | parameter_list
8565                                     ;
8566     parameter_list                   : LETTER
8567                                     | define_list ',' LETTER
8568                                     ;
8569     opt_auto_define_list              : /* empty */
8570                                     | Auto define_list NEWLINE
8571                                     | Auto define_list ';'
8572                                     ;
8573     define_list                       : LETTER
8574                                     | LETTER '[' ']'
8575                                     | define_list ',' LETTER
8576                                     | define_list ',' LETTER '[' ']'
8577                                     ;
8578     opt_argument_list                 : /* empty */
8579                                     | argument_list
8580                                     ;
8581     argument_list                     : expression
8582                                     | LETTER '[' ']' ',' argument_list"
8583                                     ;

```

```

8584     relational_expression : expression
8585                             | expression REL_OP expression
8586                             ;
8587     return_expression      : /* empty */
8588                             | expression
8589                             ;
8590     expression             : named_expression
8591                             | NUMBER
8592                             | '(' expression ')'
8593                             | LETTER '(' opt_argument_list ')'
8594                             | '-' expression
8595                             | expression '+' expression
8596                             | expression '-' expression
8597                             | expression MUL_OP expression
8598                             | expression '^' expression
8599                             | INCR_DECR named_expression
8600                             | named_expression INCR_DECR
8601                             | named_expression ASSIGN_OP expression
8602                             | Length '(' expression ')'
8603                             | Sqrt '(' expression ')'
8604                             | Scale '(' expression ')'
8605                             ;
8606     named_expression       : LETTER
8607                             | LETTER '[' expression ']'
8608                             | Scale
8609                             | Ibase
8610                             | Obase
8611                             ;

```

8612 Lexical Conventions in bc

8613 The lexical conventions for *bc* programs, with respect to the preceding grammar, shall be as
 8614 follows:

- 8615 1. Except as noted, *bc* shall recognize the longest possible token or delimiter beginning at a
 8616 given point.
- 8617 2. A comment shall consist of any characters beginning with the two adjacent characters
 8618 `"/*"` and terminated by the next occurrence of the two adjacent characters `"*/"`.
 8619 Comments shall have no effect except to delimit lexical tokens.
- 8620 3. The `<newline>` character shall be recognized as the token **NEWLINE**.
- 8621 4. The token **STRING** shall represent a string constant; it shall consist of any characters
 8622 beginning with the double-quote character (`'"`) and terminated by another occurrence of
 8623 the double-quote character. The value of the string is the sequence of all characters
 8624 between, but not including, the two double-quote characters. All characters shall be taken
 8625 literally from the input, and there is no way to specify a string containing a double-quote
 8626 character. The length of the value of each string shall be limited to `{BC_STRING_MAX}`
 8627 bytes.
- 8628 5. A `<blank>` character shall have no effect except as an ordinary character if it appears
 8629 within a **STRING** token, or to delimit a lexical token other than **STRING**.

- 8630 6. The combination of a backslash character immediately followed by a <newline> character
8631 shall have no effect other than to delimit lexical tokens with the following exceptions:
- 8632 • It shall be interpreted as the character sequence "\<newline>" in **STRING** tokens.
 - 8633 • It shall be ignored as part of a multi-line **NUMBER** token.
- 8634 7. The token **NUMBER** shall represent a numeric constant. It shall be recognized by the
8635 following grammar:
- ```

8636 NUMBER : integer
8637 | '.' integer
8638 | integer '.'
8639 | integer '.' integer
8640 ;

8641 integer : digit
8642 | integer digit
8643 ;

8644 digit : 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7
8645 | 8 | 9 | A | B | C | D | E | F
8646 ;

```
- 8647 8. The value of a **NUMBER** token shall be interpreted as a numeral in the base specified by  
8648 the value of the internal register **ibase** (described below). Each of the **digit** characters shall  
8649 have the value from 0 to 15 in the order listed here, and the period character shall represent  
8650 the radix point. The behavior is undefined if digits greater than or equal to the value of  
8651 **ibase** appear in the token. However, note the exception for single-digit values being  
8652 assigned to **ibase** and **obase** themselves, in **Operations in bc** on page 233.
- 8653 9. The following keywords shall be recognized as tokens:
- |      |               |              |               |               |              |
|------|---------------|--------------|---------------|---------------|--------------|
| 8654 | <b>auto</b>   | <b>ibase</b> | <b>length</b> | <b>return</b> | <b>while</b> |
| 8655 | <b>break</b>  | <b>if</b>    | <b>obase</b>  | <b>scale</b>  |              |
| 8656 | <b>define</b> | <b>for</b>   | <b>quit</b>   | <b>sqrt</b>   |              |
- 8657 10. Any of the following characters occurring anywhere except within a keyword shall be  
8658 recognized as the token **LETTER**:
- 8659 a b c d e f g h i j k l m n o p q r s t u v w x y z
- 8660 11. The following single-character and two-character sequences shall be recognized as the  
8661 token **ASSIGN\_OP**:
- 8662 = += -= \*= /= %= ^=
- 8663 12. If an '=' character, as the beginning of a token, is followed by a '-' character with no  
8664 intervening delimiter, the behavior is undefined.
- 8665 13. The following single-characters shall be recognized as the token **MUL\_OP**:
- 8666 \* / %
- 8667 14. The following single-character and two-character sequences shall be recognized as the  
8668 token **REL\_OP**:
- 8669 == <= >= != < >
- 8670 15. The following two-character sequences shall be recognized as the token **INCR\_DECR**:

- 8671            ++    --
- 8672            16. The following single characters shall be recognized as tokens whose names are the
- 8673            character:
- 8674            <newline> ( ) , + - ; [ ] ^ { }
- 8675            17. The token **EOF** is returned when the end of input is reached.

## 8676            Operations in bc

8677            There are three kinds of identifiers: ordinary identifiers, array identifiers, and function

8678            identifiers. All three types consist of single lowercase letters. Array identifiers shall be followed

8679            by square brackets ("[]"). An array subscript is required except in an argument or auto list.

8680            Arrays are singly dimensioned and can contain up to {BC\_DIM\_MAX} elements. Indexing shall

8681            begin at zero so an array is indexed from 0 to {BC\_DIM\_MAX}-1. Subscripts shall be truncated

8682            to integers. The application shall ensure that function identifiers are followed by parentheses,

8683            possibly enclosing arguments. The three types of identifiers do not conflict.

8684            The following table summarizes the rules for precedence and associativity of all operators.

8685            Operators on the same line shall have the same precedence; rows are in order of decreasing

8686            precedence.

8687            **Table 4-3 Operators in bc**

| Operator                   | Associativity |
|----------------------------|---------------|
| ++, --                     | N/A           |
| unary -                    | N/A           |
| ^                          | Right to left |
| *, /, %                    | Left to right |
| +, binary -                | Left to right |
| =, +=, -=, *=, /=, % =, ^= | Right to left |
| ==, <=, >=, !=, <, >       | None          |

8696            Each expression or named expression has a *scale*, which is the number of decimal digits that

8697            shall be maintained as the fractional portion of the expression.

8698            *Named expressions* are places where values are stored. Named expressions shall be valid on the

8699            left side of an assignment. The value of a named expression shall be the value stored in the place

8700            named. Simple identifiers and array elements are named expressions; they have an initial value

8701            of zero and an initial scale of zero.

8702            The internal registers **scale**, **ibase**, and **obase** are all named expressions. The scale of an

8703            expression consisting of the name of one of these registers shall be zero; values assigned to any

8704            of these registers are truncated to integers. The **scale** register shall contain a global value used in

8705            computing the scale of expressions (as described below). The value of the register **scale** is

8706            limited to  $0 \leq \text{scale} \leq \{\text{BC\_SCALE\_MAX}\}$  and shall have a default value of zero. The **ibase** and

8707            **obase** registers are the input and output number radix, respectively. The value of **ibase** shall be

8708            limited to:

8709             $2 \leq \text{ibase} \leq 16$

8710            The value of **obase** shall be limited to:

8711             $2 \leq \text{obase} \leq \{\text{BC\_BASE\_MAX}\}$

8712            When either **ibase** or **obase** is assigned a single **digit** value from the list in **Lexical Conventions**

8713            in **bc** on page 231, the value shall be assumed in hexadecimal. (For example, **ibase=A** sets to

base ten, regardless of the current **ibase** value.) Otherwise, the behavior is undefined when digits greater than or equal to the value of **ibase** appear in the input. Both **ibase** and **obase** shall have initial values of 10.

Internal computations shall be conducted as if in decimal, regardless of the input and output bases, to the specified number of decimal digits. When an exact result is not achieved, (for example, **scale**=0; 3.2/1) the result shall be truncated.

For all values of **obase** specified by this volume of IEEE Std. 1003.1-200x, *bc* shall output numeric values by performing each of the following steps in order:

1. If the value is less than zero, a hyphen ('-') character shall be output.
2. One of the following is output, depending on the numerical value:
  - If the absolute value of the numerical value is greater than or equal to one, the integer portion of the value shall be output as a series of digits appropriate to **obase** (as described below) most significant digit first. The most significant non-zero digit shall be output next, followed by each successively less significant digit.
  - If the absolute value of the numerical value is less than one but greater than zero and the scale of the numerical value is greater than zero, it is unspecified whether the character 0 is output.
  - If the numerical value is zero, the character 0 shall be output.
3. If the scale of the value is greater than zero and the numeric value is not zero, a period character shall be output, followed by a series of digits appropriate to **obase** (as described below) representing the most significant portion of the fractional part of the value. If *s* represents the scale of the value being output, the number of digits output shall be *s* if **obase** is 10, less than or equal to *s* if **obase** is greater than 10, or greater than or equal to *s* if **obase** is less than 10. For **obase** values other than 10, this should be the number of digits needed to represent a precision of  $10^s$ .

For **obase** values from 2 to 16, valid digits are the first **obase** of the single characters:

0 1 2 3 4 5 6 7 8 9 A B C D E F

which represent the values zero to 15, inclusive, respectively.

For bases greater than 16, each digit shall be written as a separate multi-digit decimal number. Each digit except the most significant fractional digit shall be preceded by a single <space> character. For bases from 17 to 100, *bc* shall write two-digit decimal numbers; for bases from 101 to 1 000, three-digit decimal strings, and so on. For example, the decimal number 1 024 in base 25 would be written as:

Δ01Δ15Δ24

in base 125, as:

Δ008Δ024

Very large numbers shall be split across lines with 70 characters per line in the POSIX locale; other locales may split at different character boundaries. Lines that are continued shall end with a backslash ('\').

A function call shall consist of a function name followed by parentheses containing a comma-separated list of expressions, which are the function arguments. A whole array passed as an argument shall be specified by the array name followed by empty square brackets. All function arguments shall be passed by value. As a result, changes made to the formal parameters shall have no effect on the actual arguments. If the function terminates by executing a **return**

8758 statement, the value of the function shall be the value of the expression in the parentheses of the  
 8759 **return** statement or shall be zero if no expression is provided or if there is no **return** statement.

8760 The result of **sqrt**(*expression*) shall be the square root of the expression. The result shall be  
 8761 truncated in the least significant decimal place. The scale of the result shall be the scale of the  
 8762 expression or the value of **scale**, whichever is larger.

8763 The result of **length**(*expression*) shall be the total number of significant decimal digits in the  
 8764 expression. The scale of the result shall be zero.

8765 The result of **scale**(*expression*) shall be the scale of the expression. The scale of the result shall be  
 8766 zero.

8767 A numeric constant shall be an expression. The scale shall be the number of digits that follow the  
 8768 radix point in the input representing the constant, or zero if no radix point appears.

8769 The sequence ( *expression* ) shall be an expression with the same value and scale as *expression*.  
 8770 The parentheses can be used to alter the normal precedence.

8771 The semantics of the unary and binary operators are as follows:

8772 **-expression**  
 8773 The result shall be the negative of the *expression*. The scale of the result shall be the scale of  
 8774 *expression*.

8775 The unary increment and decrement operators shall not modify the scale of the named  
 8776 expression upon which they operate. The scale of the result shall be the scale of that named  
 8777 expression.

8778 **++named-expression**  
 8779 The named expression shall be incremented by one. The result shall be the value of the  
 8780 named expression after incrementing.

8781 **--named-expression**  
 8782 The named expression shall be decremented by one. The result shall be the value of the  
 8783 named expression after decrementing.

8784 **named-expression++**  
 8785 The named expression shall be incremented by one. The result shall be the value of the  
 8786 named expression before incrementing.

8787 **named-expression--**  
 8788 The named expression shall be decremented by one. The result shall be the value of the  
 8789 named expression before decrementing.

8790 The exponentiation operator, circumflex ( **^** ), shall bind right to left.

8791 **expression^expression**  
 8792 The result shall be the first *expression* raised to the power of the second *expression*. If the  
 8793 second expression is not an integer, the behavior is undefined. If *a* is the scale of the left  
 8794 expression and *b* is the absolute value of the right expression, the scale of the result shall be:  
 8795 if *b* >= 0 min(*a* \* *b*, max(scale, *a*)) if *b* < 0 scale

8796 The multiplicative operators ( **\***, **/**, **%** ) shall bind left to right.

8797 **expression\*expression**  
 8798 The result shall be the product of the two expressions. If *a* and *b* are the scales of the two  
 8799 expressions, then the scale of the result shall be:

8800 `min(a+b,max(scale,a,b))`

8801 *expression/expression*

8802 The result shall be the quotient of the two expressions. The scale of the result shall be the  
8803 value of **scale**.

8804 *expression%expression*

8805 For expressions *a* and *b*, *a%b* shall be evaluated equivalent to the steps:

8806 1. Compute *a/b* to current scale.

8807 2. Use the result to compute:

8808 `a - (a / b) * b`

8809 to scale:

8810 `max(scale + scale(b), scale(a))`

8811 The scale of the result shall be:

8812 `max(scale + scale(b), scale(a))`

8813 When **scale** is zero, the '`%`' operator is the mathematical remainder operator.

8814 The additive operators ('+', '-') shall bind left to right.

8815 *expression+expression*

8816 The result shall be the sum of the two expressions. The scale of the result shall be the  
8817 maximum of the scales of the expressions.

8818 *expression-expression*

8819 The result shall be the difference of the two expressions. The scale of the result shall be the  
8820 maximum of the scales of the expressions.

8821 The assignment operators ('=', '+=', '-=', '\*=', '/=', '%=', '^=') shall bind right to left.

8822 *named-expression=expression*

8823 This expression results in assigning the value of the expression on the right to the named  
8824 expression on the left. The scale of both the named expression and the result shall be the  
8825 scale of *expression*.

8826 The compound assignment forms:

8827 *named-expression <operator>= expression*

8828 shall be equivalent to:

8829 *named-expression=named-expression <operator> expression*

8830 except that the *named-expression* shall be evaluated only once.

8831 Unlike all other operators, the relational operators ('<', '>', '<=', '>=', '==', '!=') shall be  
8832 only valid as the object of an **if**, **while**, or inside a **for** statement.

8833 *expression1<expression2*

8834 The relation shall be true if the value of *expression1* is strictly less than the value of  
8835 *expression2*.

8836 *expression1>expression2*

8837 The relation shall be true if the value of *expression1* is strictly greater than the value of  
8838 *expression2*.



8839 *expression1* <= *expression2*  
 8840 The relation shall be true if the value of *expression1* is less than or equal to the value of  
 8841 *expression2*.

8842 *expression1* >= *expression2*  
 8843 The relation shall be true if the value of *expression1* is greater than or equal to the value of  
 8844 *expression2*.

8845 *expression1* = *expression2*  
 8846 The relation shall be true if the values of *expression1* and *expression2* are equal.

8847 *expression1* != *expression2*  
 8848 The relation shall be true if the values of *expression1* and *expression2* are unequal.

8849 There are only two storage classes in *bc*, global and automatic (local). Only identifiers that are  
 8850 local to a function need be declared with the **auto** command. The arguments to a function shall  
 8851 be local to the function. All other identifiers are assumed to be global and available to all  
 8852 functions. All identifiers, global and local, have initial values of zero. Identifiers declared as auto  
 8853 shall be allocated on entry to the function and released on returning from the function. They  
 8854 therefore do not retain values between function calls. Auto arrays shall be specified by the array  
 8855 name followed by empty square brackets. On entry to a function, the old values of the names  
 8856 that appear as parameters and as automatic variables shall be pushed onto a stack. Until the  
 8857 function returns, reference to these names shall refer only to the new values.

8858 References to any of these names from other functions that are called from this function also  
 8859 refer to the new value until one of those functions uses the same name for a local variable.

8860 When a statement is an expression, unless the main operator is an assignment, execution of the  
 8861 statement shall write the value of the expression followed by a <newline> character.

8862 When a statement is a string, execution of the statement shall write the value of the string.

8863 Statements separated by semicolons or <newline> characters shall be executed sequentially. In  
 8864 an interactive invocation of *bc*, each time a <newline> character is read that satisfies the  
 8865 grammatical production:

8866 `input_item : semicolon_list NEWLINE`

8867 the sequential list of statements making up the **semicolon\_list** shall be executed immediately  
 8868 and any output produced by that execution shall be written without any delay due to buffering.

8869 In an **if** statement (**if**(*relation*) *statement*), the *statement* shall be executed if the relation is true.

8870 The **while** statement (**while**(*relation*) *statement*) implements a loop in which the *relation* is tested;  
 8871 each time the *relation* is true, the *statement* shall be executed and the *relation* retested. When the  
 8872 *relation* is false, execution shall resume after *statement*.

8873 A **for** statement(**for**(*expression*; *relation*; *expression*) *statement*) shall be the same as:

8874 *first-expression*  
 8875 while (*relation*) {  
 8876     *statement*  
 8877     *last-expression*  
 8878 }

8879 The application shall ensure that all three expressions are present.

8880 The **break** statement shall cause termination of a **for** or **while** statement.

8881 The **auto** statement (**auto** *identifier* [*identifier*] ...) shall cause the values of the identifiers to be  
 8882 pushed down. The identifiers can be ordinary identifiers or array identifiers. Array identifiers

8883 shall be specified by following the array name by empty square brackets. The application shall  
 8884 ensure that the **auto** statement is the first statement in a function definition.

8885 A **define** statement:

```
8886 define LETTER (opt_parameter_list) {
8887 opt_auto_define_list
8888 statement_list
8889 }
```

8890 defines a function named **LETTER**. If a function named **LETTER** was previously defined, the  
 8891 **define** statement shall replace the previous definition. The expression:

```
8892 LETTER (opt_argument_list)
```

8893 shall invoke the function named **LETTER**. The behavior is undefined if the number of  
 8894 arguments in the invocation does not match the number of parameters in the definition.  
 8895 Functions shall be defined before they are invoked. A function shall be considered to be defined  
 8896 within its own body, so recursive calls are valid. The values of numeric constants within a  
 8897 function shall be interpreted in the base specified by the value of the **ibase** register when the  
 8898 function is invoked.

8899 The **return** statements (**return** and **return(expression)**) shall cause termination of a function,  
 8900 popping of its auto variables, and specification of the result of the function. The first form shall  
 8901 be equivalent to **return(0)**. The value and scale of the result returned by the function shall be the  
 8902 value and scale of the expression returned.

8903 The **quit** statement (**quit**) shall stop execution of a *bc* program at the point where the statement  
 8904 occurs in the input, even if it occurs in a function definition, or in an **if**, **for**, or **while** statement.

8905 The following functions shall be defined when the **-l** option is specified:

```
8906 s(expression)
8907 Sine of argument in radians.
```

```
8908 c(expression)
8909 Cosine of argument in radians.
```

```
8910 a(expression)
8911 Arctangent of argument.
```

```
8912 l(expression)
8913 Natural logarithm of argument.
```

```
8914 e(expression)
8915 Exponential function of argument.
```

```
8916 j(expression, expression)
8917 Bessel function of integer order.
```

8918 The scale of the result returned by these functions shall be the value of the **scale** register at the  
 8919 time the function is invoked. The value of the **scale** register after these functions have completed  
 8920 their execution shall be the same value it had upon invocation. The behavior is undefined if any  
 8921 of these functions is invoked with an argument outside the domain of the mathematical  
 8922 function.

## 8923 EXIT STATUS

8924 The following exit values shall be returned:

8925 0 All input files were processed successfully.

8926            *unspecified*    An error occurred.

## 8927 CONSEQUENCES OF ERRORS

8928            If any *file* operand is specified and the named file cannot be accessed, *bc* shall write a diagnostic message to standard error and terminate without any further action.

8930            In an interactive invocation of *bc*, the utility should print an error message and recover following any error in the input. In a non-interactive invocation of *bc*, invalid input causes undefined behavior.

## 8933 APPLICATION USAGE

8934            Automatic variables in *bc* do not work in exactly the same way as in either C or PL/1.

8935            For historical reasons, the exit status from *bc* cannot be relied upon to indicate that an error has occurred. Returning zero after an error is possible. Therefore, *bc* should be used primarily by interactive users (who can react to error messages) or by application programs that can somehow validate the answers returned as not including error messages.

8939            The *bc* utility always uses the period ( `'.'` ) character to represent a radix point, regardless of any decimal-point character specified as part of the current locale. In languages like C or *awk*, the period character is used in program source, so it can be portable and unambiguous, while the locale-specific character is used in input and output. Because there is no distinction between source and input in *bc*, this arrangement would not be possible. Using the locale-specific character in *bc*'s input would introduce ambiguities into the language; consider the following example in a locale with a comma as the decimal-point character:

```
8946 define f(a,b) {
8947 ...
8948 }
8949 ...
8950 f(1,2,3)
```

8951            Because of such ambiguities, the period character is used in input. Having input follow different conventions from output would be confusing in either pipeline usage or interactive usage, so the period is also used in output.

## 8954 EXAMPLES

8955            In the shell, the following assigns an approximation of the first ten digits of ' $\pi$ ' to the variable *x*:

```
8957 x=$(printf "%s\n" 'scale = 10; 104348/33215' | bc)
```

8958            The following *bc* program prints the same approximation of ' $\pi$ ', with a label, to standard output:

```
8960 scale = 10
8961 "pi equals "
8962 104348 / 33215
```

8963            The following defines a function to compute an approximate value of the exponential function (note that such a function is predefined if the `-l` option is specified):

```
8965 scale = 20
8966 define e(x){
8967 auto a, b, c, i, s
8968 a = 1
8969 b = 1
8970 s = 1
```

```

8971 for (i = 1; i == 1; i++){
8972 a = a*x
8973 b = b*i
8974 c = a/b
8975 if (c == 0) {
8976 return(s)
8977 }
8978 s = s+c
8979 }
8980 }

```

8981 The following prints approximate values of the exponential function of the first ten integers:

```

8982 for (i = 1; i <= 10; ++i) {
8983 e(i)
8984 }

```

#### 8985 RATIONALE

8986 The *bc* utility is implemented historically as a front-end processor for *dc*; *dc* was not selected to  
 8987 be part of this volume of IEEE Std. 1003.1-200x because *bc* was thought to have a more intuitive  
 8988 programmatic interface. Current implementations that implement *bc* using *dc* are expected to be  
 8989 compliant.

8990 The exit status for error conditions has been left unspecified for several reasons:

- 8991 • The *bc* utility is used in both interactive and non-interactive situations. Different exit codes  
 8992 may be appropriate for the two uses.
- 8993 • It is unclear when a non-zero exit should be given; divide-by-zero, undefined functions, and  
 8994 syntax errors are all possibilities.
- 8995 • It is not clear what utility the exit status has.
- 8996 • In the 4.3 BSD, System V, and Ninth Edition implementations, *bc* works in conjunction with  
 8997 *dc*. The *dc* utility is the parent, *bc* is the child. This was done to cleanly terminate *bc* if *dc*  
 8998 aborted.

8999 The decision to have *bc* exit upon encountering an inaccessible input file is based on the belief  
 9000 that *bc file1 file2* is used most often when at least *file1* contains data/function  
 9001 declarations/initializations. Having *bc* continue with prerequisite files missing is probably not  
 9002 useful. There is no implication in the CONSEQUENCES OF ERRORS section that *bc* must check  
 9003 all its files for accessibility before opening any of them.

9004 There was considerable debate on the appropriateness of the language accepted by *bc*. Several  
 9005 reviewers preferred to see either a pure subset of the C language or some changes to make the  
 9006 language more compatible with C. While the *bc* language has some obvious similarities to C, it  
 9007 has never claimed to be compatible with any version of C. An interpreter for a subset of C might  
 9008 be a very worthwhile utility, and it could potentially make *bc* obsolete. However, no such utility  
 9009 is known in historical practice, and it was not within the scope of this volume of  
 9010 IEEE Std. 1003.1-200x to define such a language and utility. If and when they are defined, it may  
 9011 be appropriate to include them in a future version of this volume of IEEE Std. 1003.1-200x. This  
 9012 left the following alternatives:

- 9013 1. Exclude any calculator language from this volume of IEEE Std. 1003.1-200x.

9014 The consensus of the standard developers was that a simple programmatic calculator  
 9015 language is very useful for both applications and interactive users. The only arguments for  
 9016 excluding any calculator were that it would become obsolete if and when a C-compatible

one emerged, or that the absence would encourage the development of such a C-compatible one. These arguments did not sufficiently address the needs of current application writers.

2. Standardize the historical *dc*, possibly with minor modifications.

The consensus of the standard developers was that *dc* is a fundamentally less usable language and that that would be far too severe a penalty for avoiding the issue of being similar to but incompatible with C.

3. Standardize the historical *bc*, possibly with minor modifications.

This was the approach taken. Most of the proponents of changing the language would not have been satisfied until most or all of the incompatibilities with C were resolved. Since most of the changes considered most desirable would break historical applications and require significant modification to historical implementations, almost no modifications were made. The one significant modification that was made was the replacement of the historical *bc* assignment operators "=", and so on, with the more modern "+=", and so on. The older versions are considered to be fundamentally flawed because of the lexical ambiguity in uses like *a=-1*.

In order to permit implementations to deal with backwards compatibility as they see fit, the behavior of this one ambiguous construct was made undefined. (At least three implementations have been known to support this change already, so the degree of change involved should not be great.)

The '%' operator is the mathematical remainder operator when **scale** is zero. The behavior of this operator for other values of **scale** is from historical implementations of *bc*, and has been maintained for the sake of historical applications despite its non-intuitive nature.

Historical implementations permit setting **ibase** and **obase** to a broader range of values. This includes values less than 2, which were not seen as sufficiently useful to standardize. These implementations do not interpret input properly for values of **ibase** that are greater than 16. This is because numeric constants are recognized syntactically, rather than lexically, as described in this volume of IEEE Std. 1003.1-200x. They are built from lexical tokens of single hexadecimal digits and periods. Since <blank>s between tokens are not visible at the syntactic level, it is not possible to recognize the multi-digit "digits" used in the higher bases properly. The ability to recognize input in these bases was not considered useful enough to require modifying these implementations. Note that the recognition of numeric constants at the syntactic level is not a problem with conformance to this volume of IEEE Std. 1003.1-200x, as it does not impact the behavior of portable applications (and correct *bc* programs). Historical implementations also accept input with all of the digits '0'-'9' and 'A'-'F' regardless of the value of **ibase**; since digits with value greater than or equal to **ibase** are not really appropriate, the behavior when they appear is undefined, except for the common case of:

```
ibase=8;
/* Process in octal base. */
...
ibase=A
/* Restore decimal base. */
```

In some historical implementations, if the expression to be written is an uninitialized array element, a leading <space> character and/or up to four leading 0 characters may be output before the character zero. This behavior is considered a bug; it is unlikely that any currently portable application relies on:

9063           echo 'b[3]' | bc

9064           returning 00000 rather than 0.

9065           Exact calculation of the number of fractional digits to output for a given value in a base other  
 9066           than 10 can be computationally expensive. Historical implementations use a faster  
 9067           approximation, and this is permitted. Note that the requirements apply only to values of **obase**  
 9068           that this volume of IEEE Std. 1003.1-200x requires implementations to support (in particular, not  
 9069           to 1, 0, or negative bases, if an implementation supports them as an extension).

9070           Historical implementations of *bc* did not allow array parameters to be passed as the last  
 9071           parameter to a function. New implementations are encouraged to remove this restriction even  
 9072           though it is not required by the grammar.

9073   **FUTURE DIRECTIONS**

9074           None.

9075   **SEE ALSO**

9076           *awk*

9077   **CHANGE HISTORY**

9078           First released in Issue 4.

9079   **Issue 5**

9080           FUTURE DIRECTIONS section added.

9081   **Issue 6**

9082           Updated to align with the IEEE P1003.2b draft standard, which included resolution of several  
 9083           interpretations of the ISO POSIX-2: 1993 standard.

9084           The normative text is reworded to avoid use of the term “must” for application requirements.

9085 **NAME**

9086           bg — run jobs in the background

9087 **SYNOPSIS**9088 UP       bg [*job\_id* ...]  
90899090 **DESCRIPTION**

9091       If job control is enabled (see the description of *set -m*), the *bg* utility shall resume suspended jobs  
 9092       from the current environment (see Section 2.12 on page 90) by running them as background jobs.  
 9093       If the job specified by *job\_id* is already a running background job, the *bg* utility shall have no  
 9094       effect and shall exit successfully.

9095       Using *bg* to place a job into the background shall cause its process ID to become “known in the  
 9096       current shell execution environment”, as if it had been started as an asynchronous list; see  
 9097       Section 2.9.3.1 on page 74.

9098 **OPTIONS**

9099       None.

9100 **OPERANDS**

9101       The following operand shall be supported:

9102       *job\_id*       Specify the job to be resumed as a background job. If no *job\_id* operand is given,  
 9103       the most recently suspended job shall be used. The format of *job\_id* is described in  
 9104       the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 3.207,  
 9105       Job Control Job ID.

9106 **STDIN**

9107       Not used.

9108 **INPUT FILES**

9109       None.

9110 **ENVIRONMENT VARIABLES**9111       The following environment variables shall affect the execution of *bg*:

9112       *LANG*       Provide a default value for the internationalization variables that are unset or null.  
 9113       If *LANG* is unset or null, the corresponding value from the implementation-  
 9114       dependent default locale shall be used. If any of the internationalization variables  
 9115       contains an invalid setting, the utility shall behave as if none of the variables had  
 9116       been defined.

9117       *LC\_ALL*       If set to a non-empty string value, override the values of all the other  
 9118       internationalization variables.

9119       *LC\_CTYPE*   Determine the locale for the interpretation of sequences of bytes of text data as  
 9120       characters (for example, single-byte as opposed to multi-byte characters in  
 9121       arguments).

9122       *LC\_MESSAGES*

9123       Determine the locale that should be used to affect the format and contents of  
 9124       diagnostic messages written to standard error.

9125 XSI       *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

9126 **ASYNCHRONOUS EVENTS**

9127 Default.

9128 **STDOUT**9129 The output of *bg* shall consist of a line in the format:

9130 "[%d] %s\n", &lt;job-number&gt;, &lt;command&gt;

9131 where the fields are as follows:

9132 <job-number> A number that can be used to identify the job to the *wait*, *fg*, and *kill* utilities. Using  
9133 these utilities, the job can be identified by prefixing the job number with '% '.

9134 &lt;command&gt; The associated command that was given to the shell.

9135 **STDERR**

9136 Used only for diagnostic messages.

9137 **OUTPUT FILES**

9138 None.

9139 **EXTENDED DESCRIPTION**

9140 None.

9141 **EXIT STATUS**

9142 The following exit values shall be returned:

9143 0 Successful completion.

9144 &gt;0 An error occurred.

9145 **CONSEQUENCES OF ERRORS**9146 If job control is disabled, the *bg* utility shall exit with an error and no job shall be placed in the  
9147 background.9148 **APPLICATION USAGE**9149 A job is generally suspended by typing the SUSP character (<control>-Z on most systems); see  
9150 the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal  
9151 Interface. At that point, *bg* can put the job into the background. This is most effective when the  
9152 job is expecting no terminal input and its output has been redirected to non-terminal files. A  
9153 background job can be forced to stop when it has terminal output by issuing the command:9154 `stty tostop`

9155 A background job can be stopped with the command:

9156 `kill -s stop job ID`9157 The *bg* utility does not work as expected when it is operating in its own utility execution  
9158 environment because that environment has no suspended jobs. In the following examples:9159 ... | xargs bg  
9160 (bg)9161 each *bg* operates in a different environment and does not share its parent shell's understanding  
9162 of jobs. For this reason, *bg* is generally implemented as a shell regular built-in.9163 Application writers should note that this utility need not be provided on systems that do not  
9164 support the User Portability Utilities option.



9165 **EXAMPLES**

9166       None.

9167 **RATIONALE**

9168       The extensions to the shell specified in this volume of IEEE Std. 1003.1-200x have mostly been  
9169       based on features provided by the KornShell. The job control features provided by *bg*, *fg*, and *jobs*  
9170       are also based on the KornShell. The standard developers examined the characteristics of the C  
9171       shell versions of these utilities and found that differences exist. Despite widespread use of the C  
9172       shell, the KornShell versions were selected for this volume of IEEE Std. 1003.1-200x to maintain a  
9173       degree of uniformity with the rest of the KornShell features selected (such as the very popular  
9174       command line editing features).

9175       The *bg* utility is expected to wrap its output if the output exceeds the number of display  
9176       columns.

9177 **FUTURE DIRECTIONS**

9178       None.

9179 **SEE ALSO**9180       *fg*, *kill*, *jobs*, *wait*9181 **CHANGE HISTORY**

9182       First released in Issue 4.

9183 **Issue 6**

9184       This utility is now marked as part of the User Portability Utilities option.

9185       The JC margin marker on the SYNOPSIS is removed since support for Job Control is mandatory  
9186       in this issue. This is a FIPS requirement.

9187 **NAME**

9188       c89 — compile standard C programs

9189 **SYNOPSIS**

```
9190 CD c89 [-c][-D name[=value]]...[-E][-g][-I directory] ... [-L directory]
9191 ... [-o outfile][-O][-s][-U name]... operand ...
```

9192

9193 **DESCRIPTION**

9194       The *c89* utility is an interface to the standard C compilation system; it shall accept source code conforming to the ISO C standard. The system conceptually consists of a compiler and link editor. The files referenced by *operands* shall be compiled and linked to produce an executable file. (It is unspecified whether the linking occurs entirely within the operation of *c89*; some systems may produce objects that are not fully resolved until the file is executed.)

9199       If the *-c* option is specified, for all path name operands of the form *file.c*, the files:

9200       \$(basename *pathname* .c).o

9201       shall be created as the result of successful compilation. If the *-c* option is not specified, it is unspecified whether such *.o* files are created or deleted for the *file.c* operands.

9203       If there are no options that prevent link editing (such as *-c* or *-E*), and all operands compile and link without error, the resulting executable file shall be written according to the *-o outfile* option (if present) or to the file *a.out*.

9206       The executable file shall be created as specified in Section 1.7.1.4 on page 11, except that the file permission bits shall be set to:

9208       S\_IRWXO | S\_IRWXG | S\_IRWXU

9209       and the bits specified by the *umask* of the process shall be cleared.

9210 **OPTIONS**

9211       The *c89* utility shall conform to the System Interface Definitions volume of |  
9212       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines, except that:

- 9213       • The *-I library* operands have the format of options, but their position within a list of  
9214       operands affects the order in which libraries are searched.
- 9215       • The order of specifying the *-I* and *-L* options is significant.
- 9216       • Portable applications shall specify each option separately; that is, grouping option letters (for  
9217       example, *-cO*) need not be recognized by all implementations.

9218       The following options shall be supported:

9219       *-c*       Suppress the link-edit phase of the compilation, and do not remove any object files  
9220       that are produced.

9221       *-g*       Produce symbolic information in the object or executable files; the nature of this  
9222       information is unspecified, and may be modified by implementation-dependent  
9223       interactions with other options.

9224       *-s*       Produce object or executable files, or both, from which symbolic and other  
9225       information not required for proper execution using the *exec* family defined in the  
9226       System Interfaces volume of IEEE Std. 1003.1-200x, has been removed (stripped). If  
9227       both *-g* and *-s* options are present, the action taken is unspecified.

9228       *-o outfile*   Use the path name *outfile*, instead of the default *a.out*, for the executable file  
9229       produced. If the *-o* option is present with *-c* or *-E*, the result is unspecified.

9230        **-D** *name*[=*value*]  
 9231            Define *name* as if by a C-language **#define** directive. If no *=value* is given, a value of  
 9232            1 shall be used. The **-D** option has lower precedence than the **-U** option. That is, if  
 9233            *name* is used in both a **-U** and a **-D** option, *name* shall be undefined regardless of  
 9234            the order of the options. Additional implementation-dependent *names* may be  
 9235            provided by the compiler. Implementations shall support at least 2 048 bytes of **-D**  
 9236            definitions and 256 *names*.

9237        **-E**            Copy C-language source files to standard output, expanding all preprocessor  
 9238            directives; no compilation shall be performed. If any operand is not a text file, the  
 9239            effects are unspecified.

9240        **-I** *directory*   Change the algorithm for searching for headers whose names are not absolute path  
 9241            names to look in the directory named by the *directory* path name before looking in  
 9242            the usual places. Thus, headers whose names are enclosed in double-quotes (" ")  
 9243            shall be searched for first in the directory of the file with the **#include** line, then in  
 9244            directories named in **-I** options, and last in the usual places. For headers whose  
 9245            names are enclosed in angle brackets (" < > "), the header shall be searched for only  
 9246            in directories named in **-I** options and then in the usual places. Directories named  
 9247            in **-I** options shall be searched in the order specified. Implementations shall  
 9248            support at least ten instances of this option in a single **c89** command invocation.

9249        **-L** *directory*   Change the algorithm of searching for the libraries named in the **-l** objects to look  
 9250            in the directory named by the *directory* path name before looking in the usual  
 9251            places. Directories named in **-L** options shall be searched in the order specified.  
 9252            Implementations shall support at least ten instances of this option in a single **c89**  
 9253            command invocation. If a directory specified by a **-L** option contains files named  
 9254            **libc.a**, **libm.a**, **libl.a**, or **liby.a**, the results are unspecified.

9255        **-O**            Optimize. The nature of the optimization is unspecified.

9256        **-U** *name*       Remove any initial definition of *name*.

9257        Multiple instances of the **-D**, **-I**, **-U**, and **-L** options can be specified.

## 9258 OPERANDS

9259        An *operand* is either in the form of a path name or the form **-l** *library*. The application shall  
 9260        ensure that at least one operand of the path name form is specified. The following operands shall  
 9261        be supported:

9262        *file.c*        A C-language source file to be compiled and optionally linked. The application  
 9263        shall ensure that the operand is of this form if the **-c** option is used.

9264        *file.a*        A library of object files typically produced by the *ar* utility, and passed directly to  
 9265        the link editor. Implementations may recognize implementation-dependent  
 9266        suffixes other than **.a** as denoting object file libraries.

9267        *file.o*        An object file produced by **c89 -c** and passed directly to the link editor.  
 9268        Implementations may recognize implementation-dependent suffixes other than **.o**  
 9269        as denoting object files.

9270        The processing of other files is implementation-dependent.

9271        **-l** *library*   (The letter ell.) Search the library named:

9272            lib*library*.a

9273        A library shall be searched when its name is encountered, so the placement of a **-l**  
 9274        operand is significant. Several standard libraries can be specified in this manner, as

9275 described in the EXTENDED DESCRIPTION section. Implementations may  
 9276 recognize implementation-dependent suffixes other than **.a** as denoting libraries.

#### 9277 **STDIN**

9278 Not used.

#### 9279 **INPUT FILES**

9280 The input file shall be one of the following: a text file containing a C-language source program,  
 9281 an object file in the format produced by **c89 -c**, or a library of object files, in the format produced  
 9282 by archiving zero or more object files, using **ar**. Implementations may supply additional utilities  
 9283 that produce files in these formats. Additional input file formats are implementation-dependent.

#### 9284 **ENVIRONMENT VARIABLES**

9285 The following environment variables shall affect the execution of **c89**:

9286 **LANG** Provide a default value for the internationalization variables that are unset or null.  
 9287 If **LANG** is unset or null, the corresponding value from the implementation-  
 9288 dependent default locale shall be used. If any of the internationalization variables  
 9289 contains an invalid setting, the utility shall behave as if none of the variables had  
 9290 been defined.

9291 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 9292 internationalization variables.

9293 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 9294 characters (for example, single-byte as opposed to multi-byte characters in  
 9295 arguments and input files).

#### 9296 **LC\_MESSAGES**

9297 Determine the locale that should be used to affect the format and contents of  
 9298 diagnostic messages written to standard error.

9299 XSI **NLSPATH** Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

9300 **TMPDIR** Provide a path name that should override the default directory for temporary files,  
 9301 XSI if any. On XSI-conforming systems, provide a path name that shall override the  
 9302 default directory for temporary files, if any.

#### 9303 **ASYNCHRONOUS EVENTS**

9304 Default.

#### 9305 **STDOUT**

9306 If more than one file operand ending in **.c** (or possibly other unspecified suffixes) is given, for  
 9307 each such file:

9308 "%s:\n", <file>

9309 may be written. These messages, if written, shall precede the processing of each input file; they  
 9310 shall not be written to the standard output if they are written to the standard error, as described  
 9311 in the **STDERR** section.

9312 If the **-E** option is specified, the standard output shall be a text file that represents the results of  
 9313 the preprocessing stage of the language; it may contain extra information appropriate for  
 9314 subsequent compilation passes.

#### 9315 **STDERR**

9316 Used only for diagnostic messages. If more than one file operand ending in **.c** (or possibly other  
 9317 unspecified suffixes) is given, for each such file:

9318 "%s:\n", <file>

9319 may be written to allow identification of the diagnostic and warning messages with the  
 9320 appropriate input file. These messages, if written, shall precede the processing of each input file;  
 9321 they shall not be written to the standard error if they are written to the standard output, as  
 9322 described in the STDOUT section.

9323 This utility may produce warning messages about certain conditions that do not warrant  
 9324 returning an error (non-zero) exit value.

## 9325 OUTPUT FILES

9326 Object files or executable files or both are produced in unspecified formats.

## 9327 EXTENDED DESCRIPTION

### 9328 Standard Libraries

9329 The *c89* utility shall recognize the following **-l** operands for standard libraries:

9330 **-l c** This operand shall make visible all library functions referenced in the System  
 9331 Interfaces volume of IEEE Std. 1003.1-200x, with the possible exception of those  
 9332 functions listed as residing in **<aio.h>**, **<arpa/inet.h>**, **<math.h>**, **<mqueue.h>**,  
 9333 **<netdb.h>**, **<netinet/in.h>**, **<pthread.h>**, **<sched.h>**, **<semaphore.h>**,  
 9334 **<sys/socket.h>**, *pthread\_atfork()* in **<unistd.h>**, and those functions marked as an  
 9335 RT extension in **<sys/mman.h>** and **<time.h>**. This operand shall not be required  
 9336 to be present to cause a search of this library.

9337 **-l l** This operand shall make visible all functions required by the C-language output of  
 9338 *lex* that are not made available through the **-l c** operand.

9339 **-l pthread** This operand shall make visible all functions referenced in **<pthread.h>** and  
 9340 *pthread\_atfork()* referenced in **<unistd.h>**. An implementation may search this  
 9341 library in the absence of this operand.

9342 **-l m** This operand shall make visible all functions referenced in **<math.h>**. An  
 9343 implementation may search this library in the absence of this operand.

9344 MAN **-l rt** This operand shall make visible all functions referenced in **<aio.h>**, **<mqueue.h>**,  
 9345 **<sched.h>**, and **<semaphore.h>**, and those functions marked as an RT extension in  
 9346 **<sys/mman.h>** and **<time.h>**. An implementation may search this library in the  
 9347 absence of this operand.

9348 **-l xnet** This operand makes visible all functions referenced in **<arpa/inet.h>**, **<netdb.h>**,  
 9349 **<netinet/in.h>**, and **<sys/socket.h>**. An implementation may search this library in  
 9350 the absence of this operand.

9351 **-l y** This operand shall make visible all functions required by the C-language output of  
 9352 *yacc* that are not made available through the **-l c** operand.

9353 In the absence of options that inhibit invocation of the link editor, such as **-c** or **-E**, the *c89* utility  
 9354 shall cause the equivalent of a **-l c** operand to be passed to the link editor as the last **-l** operand,  
 9355 causing it to be searched after all other object files and libraries are loaded.

9356 It is unspecified whether the libraries **libc.a**, **libm.a**, **librt.a**, **libpthread.a**, **libl.a**, **liby.a**, or **libxnet**  
 9357 exist as regular files. The implementation may accept as **-l** operands names of objects that do  
 9358 not exist as regular files.

**External Symbols**

The C compiler and link editor shall support the significance of external symbols up to a length of at least 31 bytes; the action taken upon encountering symbols exceeding the implementation-dependent maximum symbol length is unspecified.

The compiler and link editor shall support a minimum of 511 external symbols per source or object file, and a minimum of 4095 external symbols in total. A diagnostic message shall be written to the standard output if the implementation-dependent limit is exceeded; other actions are unspecified.

**Programming Environments**

All implementations shall support one of the following programming environments as a default. Implementations may support more than one of the following programming environments. Applications can use *sysconf()* or *getconf* to determine which programming environments are supported.

**Table 4-4** Programming Environments: Type Sizes

| Programming Environment<br><i>getconf</i> Name | Bits in<br>int | Bits in<br>long | Bits in<br>pointer | Bits in<br>off_t |
|------------------------------------------------|----------------|-----------------|--------------------|------------------|
| _XBS5_ILP32_OFF32                              | 32             | 32              | 32                 | 32               |
| _XBS5_ILP32_OFFBIG                             | 32             | 32              | 32                 | ≥64              |
| _XBS5_LP64_OFF64                               | 32             | 64              | 64                 | 64               |
| _XBS5_LPBIG_OFFBIG                             | ≥32            | ≥64             | ≥64                | ≥64              |

**Notes to Reviewers**

*This section with side shading will not appear in the final copy. - Ed.*

The names of the macros above may be changed. This has been added to the issues list.

Implementations provide configuration strings for C compiler flags, linker/loader flags, and libraries for each supported environment. When an application needs to use a specific programming environment rather than the implementation default programming environment while compiling, the application shall first verify that the implementation supports the desired environment. If the desired programming environment is supported, the application shall then invoke *c89* with the appropriate C compiler flags as the first options for the compile, the appropriate linker/loader flags after any other options but before any operands, and the appropriate libraries at the end of the operands.

Portable applications shall not attempt to link together object files compiled for different programming models. Applications shall also be aware that binary data placed in shared memory or in files might not be recognized by applications built for other programming models.

Table 4-5 Programming Environments: *c89* and *cc* Arguments

| Programming Environment<br><i>getconf</i> Name | Use                                                  | <i>c89</i> and <i>cc</i> Arguments<br><i>getconf</i> Name                       |
|------------------------------------------------|------------------------------------------------------|---------------------------------------------------------------------------------|
| _XBS5_ILP32_OFF32                              | C Compiler Flags<br>Linker/Loader Flags<br>Libraries | XBS5_ILP32_OFF32_CFLAGS<br>XBS5_ILP32_OFF32_LDFLAGS<br>XBS5_ILP32_OFF32_LIBS    |
| _XBS5_ILP32_OFFBIG                             | C Compiler Flags<br>Linker/Loader Flags<br>Libraries | XBS5_ILP32_OFFBIG_CFLAGS<br>XBS5_ILP32_OFFBIG_LDFLAGS<br>XBS5_ILP32_OFFBIG_LIBS |
| _XBS5_LP64_OFF64                               | C Compiler Flags<br>Linker/Loader Flags<br>Libraries | XBS5_LP64_OFF64_CFLAGS<br>XBS5_LP64_OFF64_LDFLAGS<br>XBS5_LP64_OFF64_LIBS       |
| _XBS5_LPBIG_OFFBIG                             | C Compiler Flags<br>Linker/Loader Flags<br>Libraries | XBS5_LPBIG_OFFBIG_CFLAGS<br>XBS5_LPBIG_OFFBIG_LDFLAGS<br>XBS5_LPBIG_OFFBIG_LIBS |

**Notes to Reviewers**

*This section with side shading will not appear in the final copy. - Ed.*

The names of the macros above may be changed. This has been added to the issues list.

**EXIT STATUS**

The following exit values shall be returned:

0 Successful compilation or link edit.

>0 An error occurred.

**CONSEQUENCES OF ERRORS**

When *c89* encounters a compilation error that causes an object file not to be created, it shall write a diagnostic to standard error and continue to compile other source code operands, but it shall not perform the link phase and return a non-zero exit status. If the link edit is unsuccessful, a diagnostic message shall be written to standard error and *c89* exits with a non-zero status. A portable application shall rely on the exit status of *c89*, rather than on the existence or mode of the executable file.

**APPLICATION USAGE**

Since the *c89* utility usually creates files in the current directory during the compilation process, it is typically necessary to run the *c89* utility in a directory in which a file can be created.

On systems providing POSIX Conformance (see the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 2, Conformance), *c89* is required only with the the C-Language Development option; XSI-conformant systems always provide *c89*.

Some historical implementations have created *.o* files when *-c* is not specified and more than one source file is given. Since this area is left unspecified, the application cannot rely on *.o* files being created, but it also must be prepared for any related *.o* files that already exist being deleted at the completion of the link edit.

Some historical implementations have permitted *-L* options to be interspersed with *-I* operands on the command line. For an application to compile consistently on systems that do not behave like this, it is necessary for a portable application to supply all *-L* options before any of the *-I* options.

There is the possible implication that if a user supplies versions of the standard library functions (before they would be encountered by an implicit **-l c** or explicit **-l m**), that those versions would be used in place of the standard versions. There are various reasons this might not be true (functions defined as macros, manipulations for clean name space, and so on), so the existence of files named in the same manner as the standard libraries within the **-L** directories is explicitly stated to produce unspecified behavior.

All of the functions specified in the System Interfaces volume of IEEE Std. 1003.1-200x may be made visible by implementations when the Standard C Library is searched. Portable applications must explicitly request searching the other standard libraries when functions made visible by those libraries are used.

## EXAMPLES

1. The following usage example compiles **foo.c** and creates the executable file **foo**:

```
c89 -o foo foo.c
```

The following usage example compiles **foo.c** and creates the object file **foo.o**:

```
c89 -c foo.c
```

The following usage example compiles **foo.c** and creates the executable file **a.out**:

```
c89 foo.c
```

The following usage example compiles **foo.c**, links it with **bar.o**, and creates the executable file **a.out**. It also creates and leaves **foo.o**:

```
c89 foo.c bar.o
```

2. The following example shows how an application using threads interfaces can test for support of and use a programming environment supporting 32-bit **int**, **long**, and **pointer** types and an **off\_t** type using at least 64 bits:

```
if [$(getconf _XBS5_ILP32_OFFBIG) != "-1"]
then
 c89 $(getconf XBS5_ILP32_OFFBIG_CFLAGS) -D_XOPEN_SOURCE=500 \
 $(getconf XBS5_ILP32_OFFBIG_LDFLAGS) foo.c -o foo \
 $(getconf XBS5_ILP32_OFFBIG_LIBS) -l pthread
else
 echo ILP32_OFFBIG programming environment not supported
 exit 1
fi
```

## Notes to Reviewers

*This section with side shading will not appear in the final copy. - Ed.*

The names of the macros above may be changed. This has been added to the issues list.



3. The following examples clarify the use and interactions of **-L** options and **-l** operands.

Consider the case in which module **a.c** calls function *f()* in library **libQ.a**, and module **b.c** calls function *g()* in library **libp.a**. Assume that both libraries reside in **/a/b/c**. The command line to compile and link in the desired way is:

```
c89 -L /a/b/c main.o a.c -l Q b.c -l p
```

In this case the **-l Q** operand need only precede the first **-l p** operand, since both **libQ.a** and **libp.a** reside in the same directory.

Multiple **-L** operands can be used when library name collisions occur. Building on the previous example, suppose that the user wants to use a new **libp.a**, in **/a/a/a**, but still wants *f()* from **/a/b/c/libQ.a**:

```
c89 -L /a/a/a -L /a/b/c main.o a.c -l Q b.c -l p
```

In this example, the linker searches the **-L** options in the order specified, and finds **/a/a/a/libp.a** before **/a/b/c/libp.a** when resolving references for **b.c**. The order of the **-l** operands is still important, however.

#### RATIONALE

The name of this utility differs from the historical *cc* name. The ISO/IEC 9899:1990 standard was approved during the original development of this volume of IEEE Std. 1003.1-200x, and it is clear that POSIX must support the ISO C standard; there is no other good way of specifying a C language. The support of the ISO C standard by *c89* also mandates the ISO C standard math libraries. An alternative approach was considered: provide an option to select the type of compilation required. However, it was found that all available option letters were already in use in the various historical *cc* utilities. Thus, this name change is being used essentially as a switch. There was some temptation to use the name change as an excuse to mandate a cleaner interface (for example, to conform to the Utility Syntax Guidelines), but this was resisted; the majority of early *c89* implementations are expected to be satisfied with historical *ccs* with only minimal changes. This was decided more from the standpoint of historical applications and makefiles than for the sake of the implementors.

Note that some implementations support a more detailed model of compilation than the one described in the normative text. In this model, the following conceptual phases may exist: preprocessor, compiler, optimizer, assembler, and link editor. Such implementations may support these additional options to the *c89* utility:

**-P** Preprocess, but do not compile, the named C programs and leave the result on corresponding files suffixed **.i**.

**-S** Compile the named C programs into assembly language and leave the assembler-language output on corresponding files suffixed **.s**. No object files are created.

**[-Wc,arg1,arg2...]**

Deliver the argument(s) *argi* to phase *c* where *c* is one of [*p02a1*] indicating preprocessor, compiler, optimizer, assembler, or link editor, respectively. For example, **-Wa,-m** passes **-m** to the assembler phase.

The **-fpq** options have been excluded since they use features that are not in this volume of IEEE Std. 1003.1-200x. In specifying that *file.a* operands are *typically* produced by *ar*, it is the intention of this volume of IEEE Std. 1003.1-200x to require that object libraries produced by *ar* be usable by *c89*, but not to preclude an implementation from supplying another utility that creates object library files.

The **-l library** operand must be capable of being interspersed with file name operands so that the order in which libraries are searched by the link editor can be specified.

9519 The search algorithm for `-I directory` states that the directory of the file with the `#include` file is  
 9520 searched first, rather than being implementation-dependent. It is believed that this reflects most  
 9521 implementations, and it disallows variations on different implementations, since this would  
 9522 make it very difficult to distribute source code in a compatible form.

9523 The `-I` options are searched in the order specified (which is left to right in English). This resolves  
 9524 the conflict of which header file is used if multiple files with the same name exist in different  
 9525 directories in the `include` path.

## 9526 **Notes to Reviewers**

9527 *This section with side shading will not appear in the final copy. - Ed.*

9528 Assuming alignment with the Single UNIX Specification, then this will be mandated in this  
 9529 issue.

9530 It is unclear whether *c89* requires such a large number of file descriptors that its requirement  
 9531 should be documented here; this volume of IEEE Std. 1003.1-200x remains silent on the issue. It  
 9532 is also noted that an undocumented feature of some C compilers is that if file descriptor 9 is  
 9533 open, a linkage trace is written to it.

9534 There is no pseudo-*printf()* specification for compile errors because no common format could be  
 9535 identified. As new C compilers are written, they are encouraged to use the following format:

```
9536 "%s: %s: %d %s\n", <compiler phase>, <file name>, \
9537 <line number>, <explanation>
```

9538 The following option proposals were considered and rejected:

- 9539 1. The `-M` option in BSD does not exist in System V and is not seen to enhance application  
 9540 portability.
- 9541 2. The `-S` option was not seen to enhance application portability and makes assumptions  
 9542 about the underlying architecture.

9543 Early proposals included a `-v` option to select a compiler version. Not only did this letter (and  
 9544 every other uppercase and lowercase letter) conflict with one historical implementation or  
 9545 another, but also there was no agreement on how many compiler versions should be defined or  
 9546 what they should mean. Another choice is to specify that the `cc` utility invoke an ISO C standard  
 9547 compiler. By specifying *c89* instead, an installation is able to link either a “common usage” or an  
 9548 ISO C standard compiler to the name `cc`. Implementors are free to select implementation-  
 9549 dependent options to select (non-portable) extensions to their existing C compiler to aid the  
 9550 transition to the ISO C standard.

9551 The `-g` and `-s` options are not specified as mutually-exclusive. Historically these two options  
 9552 have been mutually-exclusive, but because both are so loosely specified, it seemed cleaner to  
 9553 leave their interaction unspecified.

9554 The `-E` option was added because headers are not required to be separate files; these values  
 9555 could be hard-coded into the compiler or might only be accessible in a non-portable way. Hence,  
 9556 while not strictly required for application portability, this option is a practical necessity as a  
 9557 portable means for ascertaining the real effects of preprocessor statements.

9558 In BSD systems, using `-c` and `-o` in the same command causes the object module to be stored in  
 9559 the specified file. In System V, this produces an error condition. Therefore, this volume of  
 9560 IEEE Std. 1003.1-200x indicates that this is an unspecified condition.

9561 Reasonably precise specification of standard library access is required. Implementations are not  
 9562 required to have `/usr/lib/libc.a`, and so on, as many historical implementations do, but if they do

not, they are required to recognize **c**, **m**, **l**, and **y** as tokens. Libraries **l** and **y** can be empty if the library functions specified for *lex* and *yacc* are accessible through the **-l** operand. Historically, these libraries have been necessary, but they are not required for a conforming implementation.

External symbol size limits are in normative text; portable applications need to know these limits. However, the minimum maximum symbol length should be taken as a constraint on a portable application, not on an implementation, and consequently the action taken for a symbol exceeding the limit is unspecified. The minimum size for the external symbol table was added for similar reasons.

The CONSEQUENCES OF ERRORS section clearly specifies the behavior of the compiler when compilation or link-edit errors occur. The behavior of several historical implementations was examined, and the choice was made to be silent on the status of the executable, or **a.out**, file in the face of compiler or linker errors. If a linker writes the executable file, then links it on disk with *lseek()* and *write()* functions, the partially linked executable file can be left on disk and its execute bits turned off if the link edit fails. However, if the linker links the image in memory before writing the file to disk, it need not touch the executable file (if it already exists) because the link edit fails. Since both approaches are historical practice, a portable application shall rely on the exit status of **c89**, rather than on the existence or mode of the executable file.

The requirement that portable applications specify compiler options separately is to reserve the multi-character option name space for vendor-specific compiler options, which are known to exist in many historical implementations. Implementations are not required to recognize, for example, **-gc** as if it were **-g -c**; nor are they forbidden from doing so. The SYNOPSIS shows all of the options separately to highlight this requirement on applications.

Echoing file names to standard error is considered a diagnostic message because it might otherwise be difficult to associate an error message with the erring file. The text specifies either standard error or standard output for these messages because some historical practice uses standard output, but there was considerable sentiment expressed for allowing it to be on standard error instead. The rationale for using standard output is that these are not really error message headers, but a running progress report on which files have been processed. The messages are described as optional because there might be different ways of constructing the messages from the compiler that should not be precluded.

#### FUTURE DIRECTIONS

None.

#### SEE ALSO

*ar*, *getconf*, *make*, *nm*, *strip*, *umask*, the System Interfaces volume of IEEE Std. 1003.1-200x, *sysconf()*

#### CHANGE HISTORY

First released in Issue 4.

#### Issue 4, Version 2

In the **Standard Libraries** subsection, the **-l c** operand describes access to traditional interfaces if **\_XOPEN\_UNIX** is defined.

#### Issue 5

In the **Standard Libraries** subsection, the **-l pthread** and **-l rt** operands are added.

A section on the use of *sysconf()* and *getconf* for determining the supported programming environments is added.

9607 **Issue 6**

9608 This utility is now marked as part of the C-Language Development Utilities option.

9609 The Open Group corrigenda items U041/1 and U034/1 have been applied, correcting the  
9610 examples.

9611 The Open Group corrigenda item U029/3 has been applied. Leading underscores were added to  
9612 the first column of Table 4-4 on page 250 and Table 4-5 on page 251.

9613 The **-l xnet** operand is added to support networking functionality. |

9614 The following new requirements on POSIX implementations derive from alignment with the |  
9615 Single UNIX Specification:

- 9616 • The **-l rt** operand is added.
- 9617 • The section on the use of *sysconf()* and *getconf* for determining the supported programming  
9618 environments is added.

9619 The normative text is reworded to avoid use of the term “must” for application requirements. |

9620 **NAME**

9621 cal — print a calendar

9622 **SYNOPSIS**9623 xSI cal `[[month] year ]`

9624

9625 **DESCRIPTION**9626 **Notes to Reviewers**9627 *This section with side shading will not appear in the final copy. - Ed.*9628 D1, XCU, ERN 163 notes that an action is assigned to HPA regarding cleaning up of the  
9629 wording.9630 The *cal* utility shall write a Gregorian calendar to standard output. If the *year* operand is  
9631 specified, a calendar for that year shall be written. If no operands are specified, a calendar for the  
9632 current month shall be written.9633 **OPTIONS**

9634 None.

9635 **OPERANDS**

9636 The following operands shall be supported:

9637 *month* Specify the month to be displayed, represented as a decimal integer from 1  
9638 (January) to 12 (December). The default shall be the current month.9639 *year* Specify the year for which the calendar is displayed, represented as a decimal  
9640 integer from 1 to 9999. The default shall be the current year.9641 **STDIN**

9642 Not used.

9643 **INPUT FILES**

9644 None.

9645 **ENVIRONMENT VARIABLES**9646 The following environment variables shall affect the execution of *cal*:9647 *LANG* Provide a default value for the internationalization variables that are unset or null.  
9648 If *LANG* is unset or null, the corresponding value from the implementation-  
9649 dependent default locale shall be used. If any of the internationalization variables  
9650 contains an invalid setting, the utility shall behave as if none of the variables had  
9651 been defined.9652 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
9653 internationalization variables.9654 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
9655 characters (for example, single-byte as opposed to multi-byte characters in  
9656 arguments).9657 *LC\_MESSAGES*9658 Determine the locale that should be used to affect the format and contents of  
9659 diagnostic messages written to standard error, and informative messages written  
9660 to standard output.9661 *LC\_TIME* Determine the format and contents of the calendar.

9662        *NLSPATH*    Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

9663        *TZ*           Determine the timezone used to calculate the value of the current month.

9664   **ASYNCHRONOUS EVENTS**

9665        Default.

9666   **STDOUT**

9667        The standard output shall be used to display the calendar, in an unspecified format.

9668   **STDERR**

9669        Used only for diagnostic messages.

9670   **OUTPUT FILES**

9671        None.

9672   **EXTENDED DESCRIPTION**

9673        None.

9674   **EXIT STATUS**

9675        The following exit values shall be returned:

9676        0    Successful completion.

9677        >0   An error occurred.

9678   **CONSEQUENCES OF ERRORS**

9679        Default.

9680   **APPLICATION USAGE**

9681        Note that:

9682        *cal 83*

9683        refers to A.D. 83, not 1983.

9684   **EXAMPLES**

9685        None.

9686   **RATIONALE**

9687        None.

9688   **FUTURE DIRECTIONS**

9689        None.

9690   **SEE ALSO**

9691        None.

9692   **CHANGE HISTORY**

9693        First released in Issue 2.

9694   **Issue 4**

9695        Format reorganized.

9696        Internationalized environment variable support mandated.

9697 **NAME**

9698       cat — concatenate and print files

9699 **SYNOPSIS**9700       cat [-u][*file* ...]9701 **DESCRIPTION**9702       The *cat* utility reads files in sequence and writes their contents to the standard output in the  
9703       same sequence.9704 **OPTIONS**9705       The *cat* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
9706       Section 12.2, Utility Syntax Guidelines.

9707       The following option shall be supported:

9708       **-u**           Write bytes from the input file to the standard output without delay as each is  
9709                      read.9710 **OPERANDS**

9711       The following operand shall be supported:

9712       *file*           A path name of an input file. If no *file* operands are specified, the standard input is  
9713                      used. If a *file* is '-', the *cat* utility shall read from the standard input at that point  
9714                      in the sequence. The *cat* utility shall not close and reopen standard input when it is  
9715                      referenced in this way, but shall accept multiple occurrences of '-' as a *file*  
9716                      operand.9717 **STDIN**9718       The standard input is used only if no *file* operands are specified, or if a *file* operand is '-'. See  
9719       the INPUT FILES section.9720 **INPUT FILES**

9721       The input files can be any file type.

9722 **ENVIRONMENT VARIABLES**9723       The following environment variables shall affect the execution of *cat*:9724       **LANG**           Provide a default value for the internationalization variables that are unset or null.  
9725                      If *LANG* is unset or null, the corresponding value from the implementation-  
9726                      dependent default locale shall be used. If any of the internationalization variables  
9727                      contains an invalid setting, the utility shall behave as if none of the variables had  
9728                      been defined.9729       **LC\_ALL**          If set to a non-empty string value, override the values of all the other  
9730                      internationalization variables.9731       **LC\_CTYPE**       Determine the locale for the interpretation of sequences of bytes of text data as  
9732                      characters (for example, single-byte as opposed to multi-byte characters in  
9733                      arguments).9734       **LC\_MESSAGES**9735                      Determine the locale that should be used to affect the format and contents of  
9736                      diagnostic messages written to standard error.9737 **XSI**       **NLS\_PATH**   Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

9738 **ASYNCHRONOUS EVENTS**

9739 Default.

9740 **STDOUT**

9741 The standard output shall contain the sequence of bytes read from the input files. Nothing else  
9742 shall be written to the standard output.

9743 **STDERR**

9744 Used only for diagnostic messages.

9745 **OUTPUT FILES**

9746 None.

9747 **EXTENDED DESCRIPTION**

9748 None.

9749 **EXIT STATUS**

9750 The following exit values shall be returned:

9751 0 All input files were output successfully.

9752 &gt;0 An error occurred.

9753 **CONSEQUENCES OF ERRORS**

9754 Default.

9755 **APPLICATION USAGE**

9756 The **-u** option has value in prototyping non-blocking reads from FIFOs. The intent is to support  
9757 the following sequence:

9758 `mkfifo foo`9759 `cat -u foo > /dev/tty13 &`9760 `cat -u > foo`

9761 It is unspecified whether standard output is or is not buffered in the default case. This is  
9762 sometimes of interest when standard output is associated with a terminal, since buffering may  
9763 delay the output. The presence of the **-u** option guarantees that unbuffered I/O is available. It is  
9764 implementation-dependent whether the *cat* utility buffers output if the **-u** option is not  
9765 specified. Traditionally, the **-u** option is implemented using the equivalent of the *setvbuf()*  
9766 function defined in the System Interfaces volume of IEEE Std. 1003.1-200x.

9767 **EXAMPLES**

9768 The following command:

9769 `cat myfile`9770 writes the contents of the file **myfile** to standard output.

9771 The following command:

9772 `cat doc1 doc2 > doc.all`9773 concatenates the files **doc1** and **doc2** and writes the result to **doc.all**.

9774 Because of the shell language mechanism used to perform output redirection, a command such  
9775 as this:

9776 `cat doc doc.end > doc`9777 causes the original data in **doc** to be lost.

9778 The command:



9779       cat start - middle - end > file

9780       when standard input is a terminal, gets two arbitrary pieces of input from the terminal with a  
9781       single invocation of *cat*. Note, however, that if standard input is a regular file, this would be  
9782       equivalent to the command:

9783       cat start - middle /dev/null end > file

9784       because the entire contents of the file would be consumed by *cat* the first time '-' was used as a  
9785       *file* operand and an end-of-file condition would be detected immediately when '-' was  
9786       referenced the second time.

## 9787 RATIONALE

9788       Historical versions of the *cat* utility include the options *-e*, *-t*, and *-v*, which permit the ends of  
9789       lines, <tab>s, and invisible characters, respectively, to be rendered visible in the output. The  
9790       standard developers omitted these options because they provide too fine a degree of control  
9791       over what is made visible, and similar output can be obtained using a command such as:

9792       sed -n -e 's/\$/\$/' -e l pathname

9793       The *-s* option was omitted because it corresponds to different functions in BSD and System V-  
9794       based systems. The BSD *-s* option to squeeze blank lines can be accomplished by the shell script  
9795       shown in following example:

```
9796 sed -n '
9797 # Write non-empty lines.
9798 ./ {
9799 p
9800 d
9801 }
9802 # Write a single empty line, then look for more empty lines.
9803 /^$/ p
9804 # Get next line, discard the held <newline> (empty line),
9805 # and look for more empty lines.
9806 :Empty
9807 /^$/ {
9808 N
9809 s/./ /
9810 b Empty
9811 }
9812 # Write the non-empty line before going back to search
9813 # for the first in a set of empty lines.
9814 p
9815 '
```

9816       The System V *-s* option to silence error messages can be accomplished by redirecting the  
9817       standard error. Note that the BSD documentation for *cat* uses the term “blank line” to mean the  
9818       same as the POSIX “empty line”: a line consisting only of a <newline>.

9819       The BSD *-n* option was omitted because similar functionality can be obtained from the *-n*  
9820       option of the *pr* utility.

## 9821 FUTURE DIRECTIONS

9822       None.

|      |                                                 |
|------|-------------------------------------------------|
| 9823 | <b>SEE ALSO</b>                                 |
| 9824 | <i>more</i>                                     |
| 9825 | <b>CHANGE HISTORY</b>                           |
| 9826 | First released in Issue 2.                      |
| 9827 | <b>Issue 4</b>                                  |
| 9828 | Aligned with the ISO/IEC 9945-2: 1993 standard. |

## 9829 NAME

9830 cd — change the working directory

## 9831 SYNOPSIS

9832 cd [-L] [-P] [*directory*]

## 9833 MAN

9834

cd -

## 9835 DESCRIPTION

9836 The *cd* utility shall change the working directory of the current shell execution environment (see  
 9837 Section 2.12 on page 90) by executing the following steps in sequence. (In the following steps,  
 9838 the symbol **curpath** represents an intermediate value used to simplify the description of the  
 9839 algorithm used by *cd*. There is no requirement that **curpath** be made visible to the application.)

- 9840 1. If no *directory* operand is given and the *HOME* environment variable is empty or  
 9841 undefined, the default behavior is implementation-dependent and no further steps shall be  
 9842 taken.
- 9843 2. If no *directory* operand is given and the *HOME* environment variable is set to a non-empty  
 9844 value, the *cd* utility shall behave as if the directory named in the *HOME* environment  
 9845 variable was specified as the *directory* operand.
- 9846 3. Starting with the first path name in the colon-separated path names of *CDPATH* (see the  
 9847 ENVIRONMENT VARIABLES section) if the path name is non-null, test if the  
 9848 concatenation of that path name, a slash character, and the operand names a directory. If  
 9849 the path name is null, test if the concatenation of dot, a slash character, and the operand  
 9850 names a directory. In either case, if the resulting string names an existing directory, set  
 9851 **curpath** to that string and proceed to step 4. Otherwise, repeat this step with the next path  
 9852 name in *CDPATH* until all path names have been tested.
- 9853 4. If the *-P* option is in effect, the *cd* utility shall perform actions equivalent to the *chdir()*  
 9854 function, called with **curpath** as the *path* argument. If these actions succeed, the *PWD*  
 9855 environment variable shall be set to an absolute path name for the current working  
 9856 directory and shall not contain file name components that, in the context of path name  
 9857 resolution, refer to a file of type symbolic link. If there is insufficient permission on the new  
 9858 directory, or on any parent of that directory, to determine the current working directory,  
 9859 the value of the *PWD* environment variable is unspecified. If the actions equivalent to  
 9860 *chdir()* fail for any reason, the *cd* utility shall display an appropriate error message and not  
 9861 alter the *PWD* environment variable. Whether the actions equivalent to *chdir()* succeed or  
 9862 fail, no further steps shall be taken.
- 9863 5. The **curpath** value shall then be converted to canonical form as follows, considering each  
 9864 component from beginning to end, in sequence:
  - 9865 a. Dot components and any slashes that separate them from the next component shall  
 9866 be deleted.
  - 9867 b. For each dot-dot component, if there is a preceding component and it is neither root  
 9868 nor dot-dot, the preceding component, all slashes separating the preceding  
 9869 component from dot-dot, dot-dot, and all slashes separating dot-dot from the  
 9870 following component shall be deleted.
  - 9871 c. An implementation may further simplify **curpath** by removing any trailing slash  
 9872 characters that are not also leading slashes, replacing multiple non-leading  
 9873 consecutive slashes with a single slash, and replacing three or more leading  
 9874 slashes with a single slash. If, as a result of this canonicalization, the **curpath** variable  
 9875 is null, no further steps shall be taken.

9876           6. The *cd* utility shall then perform actions equivalent to the *chdir()* function called with  
 9877           **curpath** as the *path* argument. If these actions failed for any reason, the *cd* utility shall  
 9878           display an appropriate error message and no further steps shall be taken. The *PWD*  
 9879           environment variable shall be set to **curpath**.

9880   MAN    If, during the execution of the above steps, the *PWD* environment variable is changed, the  
 9881           *OLDPWD* environment variable shall also be changed to the value of the old working directory  
 9882           (that is the current working directory immediately prior to the call to *cd*).

## 9883 OPTIONS

9884           The *cd* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 9885           Section 12.2, Utility Syntax Guidelines.

9886           The following options shall be supported by the implementation:

9887           **-L**       Handle the operand dot-dot logically; symbolic link components shall not be  
 9888                       resolved before dot-dot components are processed (see steps 5. and 6. in the  
 9889                       DESCRIPTION).

9890           **-P**       Handle the operand dot-dot physically; symbolic link components shall be  
 9891                       resolved before dot-dot components are processed (see step 4. in the  
 9892                       DESCRIPTION).

9893           If both **-L** and **-P** options are specified, the last of these options shall be used and all others  
 9894           ignored. If neither **-L** nor **-P** is specified, the operand shall be handled dot-dot logically; see the  
 9895           DESCRIPTION.

## 9896 OPERANDS

9897           The following operands shall be supported:

9898           *directory*   An absolute or relative path name of the directory that shall become the new  
 9899                       working directory. The interpretation of a relative path name by *cd* depends on the  
 9900                       **-L** option and the *CDPATH* and *PWD* environment variables. If *directory* is an  
 9901                       empty string, the results are unspecified.

9902   MAN    **-**       When a hyphen is used as the operand, this is equivalent to the command:

9903           `cd "$OLDPWD" && pwd`

9904           which changes to the previous working directory and then writes its name.

## 9905 STDIN

9906           Not used.

## 9907 INPUT FILES

9908           None.

## 9909 ENVIRONMENT VARIABLES

9910           The following environment variables shall affect the execution of *cd*:

9911           *CDPATH*    A colon-separated list of path names that refer to directories. The *cd* utility shall  
 9912                       use this list in its attempt to change the directory, as described in the  
 9913                       DESCRIPTION. An empty string in place of a directory path name represents the  
 9914                       current directory. If *CDPATH* is not set, it shall be treated as if it were an empty  
 9915                       string.

9916           *HOME*      The name of the directory, used when no *directory* operand is specified.

9917           *LANG*      Provide a default value for the internationalization variables that are unset or null.  
 9918                       If *LANG* is unset or null, the corresponding value from the implementation-  
 9919                       dependent default locale shall be used. If any of the internationalization variables

9920 contains an invalid setting, the utility shall behave as if none of the variables had  
 9921 been defined.

9922 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 9923 internationalization variables.

9924 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 9925 characters (for example, single-byte as opposed to multi-byte characters in  
 9926 arguments).

9927 **LC\_MESSAGES**  
 9928 Determine the locale that should be used to affect the format and contents of  
 9929 diagnostic messages written to standard error.

9930 XSI **NLSPATH** Determine the location of message catalogs for the processing of **LC\_MESSAGES**. |  
 9931 MAN **OLDPWD** A path name of the previous working directory, used by **cd -**. |  
 9932 **PWD** This variable shall be set as specified in the DESCRIPTION. If an application sets |  
 9933 or unsets the value of **PWD**, the behavior of **cd** is unspecified. |

9934 **ASYNCHRONOUS EVENTS**  
 9935 Default.

9936 **STDOUT**  
 9937 MAN If a non-empty directory name from **CDPATH** is used, or if **cd -** is used, an absolute path name of  
 9938 the new working directory shall be written to the standard output as follows:  
 9939 "%s\n", <new directory>  
 9940 Otherwise, there shall be no output.

9941 **STDERR**  
 9942 Used only for diagnostic messages.

9943 **OUTPUT FILES**  
 9944 None.

9945 **EXTENDED DESCRIPTION**  
 9946 None.

9947 **EXIT STATUS**  
 9948 The following exit values shall be returned:  
 9949 0 The directory was successfully changed.  
 9950 >0 An error occurred.

9951 **CONSEQUENCES OF ERRORS**  
 9952 The working directory shall remain unchanged.

9953 **APPLICATION USAGE**  
 9954 Since **cd** affects the current shell execution environment, it is always provided as a shell regular  
 9955 built-in. If it is called in a subshell or separate utility execution environment, such as one of the  
 9956 following:  
 9957 (cd /tmp)  
 9958 nohup cd  
 9959 find . -exec cd {} \;  
 9960 it does not affect the working directory of the caller's environment.

9961 The user must have execute (search) permission in *directory* in order to change to it.

#### 9962 EXAMPLES

9963 None.

#### 9964 RATIONALE

9965 The use of the *CDPATH* was introduced in the System V shell. Its use is analogous to the use of  
9966 the *PATH* variable in the shell. The BSD C shell used a shell parameter *cdpath* for this purpose.

9967 A common extension when *HOME* is undefined is to get the login directory from the user  
9968 database for the invoking user. This does not occur on System V implementations.

9969 Some historical shells, such as the KornShell, took special actions when the directory name  
9970 contained a dot-dot component, selecting the logical parent of the directory, rather than the  
9971 actual parent directory; that is, it moved up one level toward the '/' in the path name,  
9972 remembering what the user typed, rather than performing the equivalent of:

9973 `chdir("../");`

9974 In such a shell, the following commands would not necessarily produce equivalent output for all  
9975 directories:

9976 `cd .. && ls           ls ..`

9977 This behavior is not permitted by default because it is not consistent with the definition of dot-  
9978 dot in most historical practice; that is, while this behavior has been optionally available in the  
9979 KornShell, other shells have historically not supported this functionality. The logical path name  
9980 is stored in the *PWD* environment variable when the *cd* utility completes and this value is used  
9981 to construct the next directory name if *cd* is invoked with the *-L* option.

#### 9982 FUTURE DIRECTIONS

9983 None.

#### 9984 SEE ALSO

9985 *pwd*, the System Interfaces volume of IEEE Std. 1003.1-200x, *chdir()*

#### 9986 CHANGE HISTORY

9987 First released in Issue 2.

#### 9988 Issue 4

9989 Aligned with the ISO/IEC 9945-2:1993 standard.

9990 Extensions added for *cd-*, *PWD*, and *OLDPWD*.

#### 9991 Issue 6

9992 The following new requirements on POSIX implementations derive from alignment with the  
9993 Single UNIX Specification:

- 9994 • The *cd-*, *PWD*, and *OLDPWD* are added.

9995 The *-L* and *-P* options are added to align with the IEEE P1003.2b draft standard. This also  
9996 includes the introduction of a new description to include the effect of these options.

9997 **NAME**9998 cflow — generate a C-language flowgraph (**DEVELOPMENT**)9999 **SYNOPSIS**

```
10000 xsi cflow [-r][-d num][-D name[=def]] ... [-i incl][-I dir] ...
10001 [-U dir] ... file ...
```

10002

10003 **DESCRIPTION**

10004 The *cflow* utility shall analyse a collection of object files or assembler, C-language, *lex* or *yacc*  
 10005 source files, and attempt to build a graph, written to standard output, charting the external  
 10006 references.

10007 **OPTIONS**

10008 The *cflow* utility shall conform to the System Interface Definitions volume of  
 10009 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines, except that the order of the **-D**, **-I**,  
 10010 and **-U** options (which are identical to their interpretation by *c89*) is significant.

10011 The following options shall be supported:

10012 **-d num** Indicate the depth at which the flowgraph is cut off. The application shall ensure  
 10013 that the argument *num* is a decimal integer. By default this is a very large number  
 10014 (typically greater than 32 000). Attempts to set the cut-off depth to a non-positive  
 10015 integer are ignored.

10016 **-i incl** Increase the number of included symbols. The *incl* option-argument is one of the  
 10017 following characters:

10018 *x* Include external and static data symbols. The default shall be to include only  
 10019 functions in the flowgraph.

10020 *\_* (Underscore) Include names that begin with an underscore. The default shall  
 10021 be to exclude these functions (and data if **-i x** is used).

10022 **-r** Reverse the caller: callee relationship, producing an inverted listing showing the  
 10023 callers of each function. The listing is also sorted in lexicographical order by callee.

10024 **OPERANDS**

10025 The following operand is supported:

10026 *file* The path name of a file for which a graph is to be generated. Files suffixed in *.l*, *.y*,  
 10027 *.c*, and *.i* shall be processed by *lex* and *yacc* and preprocessed by the *c89*  
 10028 preprocessor phase (bypassed for *.i* files) as appropriate, and then run through the  
 10029 first pass of *lint*. Files suffixed with *.s* shall be assembled and information shall be  
 10030 extracted (as in *.o* files) from the symbol table.

10031 **STDIN**

10032 Not used.

10033 **INPUT FILES**

10034 The input files shall be object files or assembler, C-language, *lex* or *yacc* source files.

10035 **ENVIRONMENT VARIABLES**

10036 The following environment variables shall affect the execution of *cflow*:

10037 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 10038 If *LANG* is unset or null, the corresponding value from the implementation-  
 10039 dependent default locale shall be used. If any of the internationalization variables  
 10040 contains an invalid setting, the utility shall behave as if none of the variables had  
 10041 been defined.

|       |                               |                                                                                                               |
|-------|-------------------------------|---------------------------------------------------------------------------------------------------------------|
| 10042 | <i>LC_ALL</i>                 | If set to a non-empty string value, override the values of all the other                                      |
| 10043 |                               | internationalization variables.                                                                               |
| 10044 | <i>LC_COLLATE</i>             |                                                                                                               |
| 10045 |                               | Determine the locale for the ordering of the output when the <i>-r</i> option is used.                        |
| 10046 | <i>LC_CTYPE</i>               | Determine the locale for the interpretation of sequences of bytes of text data as                             |
| 10047 |                               | characters (for example, single-byte as opposed to multi-byte characters in                                   |
| 10048 |                               | arguments and input files).                                                                                   |
| 10049 | <i>LC_MESSAGES</i>            |                                                                                                               |
| 10050 |                               | Determine the locale that should be used to affect the format and contents of                                 |
| 10051 |                               | diagnostic messages written to standard error.                                                                |
| 10052 | <i>NLSPATH</i>                | Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .                         |
| 10053 | <b>ASYNCHRONOUS EVENTS</b>    |                                                                                                               |
| 10054 |                               | Default.                                                                                                      |
| 10055 | <b>STDOUT</b>                 |                                                                                                               |
| 10056 |                               | The flowgraph written to standard output shall be formatted as follows:                                       |
| 10057 |                               | <code>"%d %s:%s\n", &lt;reference number&gt;, &lt;global&gt;, &lt;definition&gt;</code>                       |
| 10058 |                               | Each line of output begins with a reference (that is, line) number, followed by indentation of at             |
| 10059 |                               | least one column position per level. This is followed by the name of the global, a colon, and its             |
| 10060 |                               | definition. Normally globals are only functions not defined as an external or beginning with an               |
| 10061 |                               | underscore; see the <b>OPTIONS</b> section for the <i>-i</i> inclusion option. For information extracted from |
| 10062 |                               | C-language source, the definition consists of an abstract type declaration (for example, <b>char*</b> )       |
| 10063 |                               | and, delimited by angle brackets, the name of the source file and the line number where the                   |
| 10064 |                               | definition was found. Definitions extracted from object files indicate the file name and location             |
| 10065 |                               | counter under which the symbol appeared (for example, <i>text</i> ).                                          |
| 10066 |                               | Once a definition of a name has been written, subsequent references to that name contain only                 |
| 10067 |                               | the reference number of the line where the definition can be found. For undefined references,                 |
| 10068 |                               | only "< >" shall be written.                                                                                  |
| 10069 | <b>STDERR</b>                 |                                                                                                               |
| 10070 |                               | Used only for diagnostic messages.                                                                            |
| 10071 | <b>OUTPUT FILES</b>           |                                                                                                               |
| 10072 |                               | None.                                                                                                         |
| 10073 | <b>EXTENDED DESCRIPTION</b>   |                                                                                                               |
| 10074 |                               | None.                                                                                                         |
| 10075 | <b>EXIT STATUS</b>            |                                                                                                               |
| 10076 |                               | The following exit values shall be returned:                                                                  |
| 10077 | 0                             | Successful completion.                                                                                        |
| 10078 | >0                            | An error occurred.                                                                                            |
| 10079 | <b>CONSEQUENCES OF ERRORS</b> |                                                                                                               |
| 10080 |                               | Default.                                                                                                      |



**10081 APPLICATION USAGE**

10082 Files produced by *lex* and *yacc* cause the reordering of line number declarations, and this can  
 10083 confuse *cflow*. To obtain proper results, the input of *yacc* or *lex* must be directed to *cflow*.

**10084 EXAMPLES**

10085 Given the following in **file.c**:

```
10086 int i;
10087 main()
10088 {
10089 f();
10090 g();
10091 f();
10092 }
10093 f()
10094 {
10095 i = h();
10096 }
```

10097 The command:

```
10098 cflow -i x file.c
```

10099 produces the output:

```
10100 1 main: int(), <file.c 2>
10101 2 f: int(), <file.c 8>
10102 3 h: <>
10103 4 i: int, <file.c 1>
10104 5 g: <>
```

**10105 RATIONALE**

10106 None.

**10107 FUTURE DIRECTIONS**

10108 None.

**10109 SEE ALSO**

10110 *c89*, *lex*, *yacc*

**10111 CHANGE HISTORY**

10112 First released in Issue 2.

**10113 Issue 4**

10114 Format reorganized.

10115 Internationalized environment variable support mandated.

**10116 Issue 6**

10117 The normative text is reworded to avoid use of the term “must” for application requirements.

10118 **NAME**

10119       chgrp — change the file group ownership

10120 **SYNOPSIS**10121       chgrp -hR *group file ...*10122       chgrp -R [-H | -L | -P ] *group file ...*10123 **DESCRIPTION**10124       The *chgrp* utility shall set the group ID of the file named by each *file* operand to the group ID  
10125       specified by the *group* operand.10126       For each *file* operand, it shall perform actions equivalent to the *chown()* function defined in the  
10127       System Interfaces volume of IEEE Std. 1003.1-200x, called with the following arguments:

- 10128       • The *file* operand shall be used as the *path* argument.
- 10129       • The user ID of the file shall be used as the *owner* argument.
- 10130       • The specified group ID shall be used as the *group* argument.

10131       Unless *chgrp* is invoked by a process with appropriate privileges, the set-user-ID and set-group-  
10132       ID bits of a regular file shall be cleared upon successful completion; the set-user-ID and set-  
10133       group-ID bits of other file types may be cleared.

10134 **OPTIONS**

10135       The *chgrp* utility shall conform to the System Interface Definitions volume of  
10136       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

10137       The following options shall be supported by the implementation:

10138       **-h**       If the system supports group IDs for symbolic links, for each *file* operand that  
10139       names a file of type symbolic link, *chgrp* shall attempt to set the group ID of the  
10140       symbolic link instead of the file referenced by the symbolic link. If the system does  
10141       not support group IDs for symbolic links, for each *file* operand that names a file of  
10142       type symbolic link, *chgrp* shall do nothing more with the current file and shall go  
10143       on to any remaining files.

10144       **-H**       If the **-R** option is specified and a symbolic link referencing a file of type directory  
10145       is specified on the command line, *chgrp* shall change the group of the directory  
10146       referenced by the symbolic link and all files in the file hierarchy below it.

10147       **-L**       If the **-R** option is specified and a symbolic link referencing a file of type directory  
10148       is specified on the command line or encountered during the traversal of a file  
10149       hierarchy, *chgrp* shall change the group of the directory referenced by the symbolic  
10150       link and all files in the file hierarchy below it.

10151       **-P**       If the **-R** option is specified and a symbolic link is specified on the command line  
10152       or encountered during the traversal of a file hierarchy, *chgrp* shall change the  
10153       group ID of the symbolic link if the system supports this operation. The *chgrp*  
10154       utility shall not follow the symbolic link to any other part of the file hierarchy.

10155       **-R**       Recursively change file group IDs. For each *file* operand that names a directory,  
10156       *chgrp* shall change the group of the directory and all files in the file hierarchy below  
10157       it. Unless a **-H**, **-L**, or **-P** option is specified, it is unspecified which of these  
10158       options will be used as the default.

10159       Specifying more than one of the mutually-exclusive options **-H**, **-L**, and **-P** shall not be  
10160       considered an error. The last option specified shall determine the behavior of the utility.

**10161 OPERANDS**

10162 The following operands shall be supported:

10163 *group* A group name from the group database or a numeric group ID. Either specifies a  
 10164 group ID to be given to each file named by one of the *file* operands. If a numeric  
 10165 *group* operand exists in the group database as a group name, the group ID number  
 10166 associated with that group name is used as the group ID.

10167 *file* A path name of a file whose group ID is to be modified.

**10168 STDIN**

10169 Not used.

**10170 INPUT FILES**

10171 None.

**10172 ENVIRONMENT VARIABLES**

10173 The following environment variables shall affect the execution of *chgrp*:

10174 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 10175 If *LANG* is unset or null, the corresponding value from the implementation-  
 10176 dependent default locale shall be used. If any of the internationalization variables  
 10177 contains an invalid setting, the utility shall behave as if none of the variables had  
 10178 been defined.

10179 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 10180 internationalization variables.

10181 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 10182 characters (for example, single-byte as opposed to multi-byte characters in  
 10183 arguments).

10184 *LC\_MESSAGES*

10185 Determine the locale that should be used to affect the format and contents of  
 10186 diagnostic messages written to standard error.

10187 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

**10188 ASYNCHRONOUS EVENTS**

10189 Default.

**10190 STDOUT**

10191 Not used.

**10192 STDERR**

10193 Used only for diagnostic messages.

**10194 OUTPUT FILES**

10195 None.

**10196 EXTENDED DESCRIPTION**

10197 None.

**10198 EXIT STATUS**

10199 The following exit values shall be returned:

10200 0 The utility executed successfully and all requested changes were made.

10201 >0 An error occurred.

**10202 CONSEQUENCES OF ERRORS**

10203 If, when invoked with the **-R** option, *chgrp* attempts but fails to change the group ID of a  
 10204 particular file in a specified file hierarchy, it shall continue to process the remaining files in the  
 10205 hierarchy. If *chgrp* cannot read or search a directory within a hierarchy, it shall continue to  
 10206 process the other parts of the hierarchy that are accessible.

**10207 APPLICATION USAGE**

10208 Only the owner of a file or the user with appropriate privileges may change the owner or group  
 10209 of a file.

10210 Some systems restrict the use of *chgrp* to a user with appropriate privileges when the *group*  
 10211 specified is not the effective group ID or one of the supplementary group IDs of the calling  
 10212 process.

**10213 EXAMPLES**

10214 None.

**10215 RATIONALE**

10216 The System V and BSD versions use different exit status codes. Some implementations used the  
 10217 exit status as a count of the number of errors that occurred; this practice is unworkable since it  
 10218 can overflow the range of valid exit status values. The standard developers chose to mask these  
 10219 by specifying only 0 and >0 as exit values.

10220 The functionality of *chgrp* is described substantially through references to *chown()*. In this way,  
 10221 there is no duplication of effort required for describing the interactions of permissions, multiple  
 10222 groups, and so on.

**10223 FUTURE DIRECTIONS**

10224 None.

**10225 SEE ALSO**

10226 *chmod*, *chown*, the System Interfaces volume of IEEE Std. 1003.1-200x, *chown()*

**10227 CHANGE HISTORY**

10228 First released in Issue 2.

**10229 Issue 4**

10230 Aligned with the ISO/IEC 9945-2:1993 standard.

**10231 Issue 6**

10232 New options **-H**, **-L**, and **-P** are added to align with the IEEE P1003.2b draft standard. These  
 10233 options affect the processing of symbolic links.

10234 **NAME**

10235           chmod — change the file modes

10236 **SYNOPSIS**10237           chmod [-R] *mode file ...*10238 **DESCRIPTION**10239           The *chmod* utility shall change any or all of the file mode bits of the file named by each *file*  
10240           operand in the way specified by the *mode* operand.10241           It is implementation-dependent whether and how the *chmod* utility affects any alternate or  
10242           additional file access control mechanism (see the System Interface Definitions volume of  
10243           IEEE Std. 1003.1-200x, Section 4.1, File Access Permissions) being used for the specified file.10244           Only a process whose effective user ID matches the user ID of the file, or a process with the  
10245           appropriate privileges, shall be permitted to change the file mode bits of a file.10246 **OPTIONS**10247           The *chmod* utility shall conform to the System Interface Definitions volume of  
10248           IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

10249           The following option shall be supported:

10250           **-R**           Recursively change file mode bits. For each *file* operand that names a directory,  
10251           *chmod* shall change the file mode bits of the directory and all files in the file  
10252           hierarchy below it.10253 **OPERANDS**

10254           The following operands shall be supported:

10255           *mode*           Represents the change to be made to the file mode bits of each file named by one of  
10256           the *file* operands; see the EXTENDED DESCRIPTION section.10257           *file*           A path name of a file whose file mode bits shall be modified.10258 **STDIN**

10259           Not used.

10260 **INPUT FILES**

10261           None.

10262 **ENVIRONMENT VARIABLES**10263           The following environment variables shall affect the execution of *chmod*:10264           **LANG**           Provide a default value for the internationalization variables that are unset or null.  
10265           If **LANG** is unset or null, the corresponding value from the implementation-  
10266           dependent default locale shall be used. If any of the internationalization variables  
10267           contains an invalid setting, the utility shall behave as if none of the variables had  
10268           been defined.10269           **LC\_ALL**          If set to a non-empty string value, override the values of all the other  
10270           internationalization variables.10271           **LC\_CTYPE**       Determine the locale for the interpretation of sequences of bytes of text data as  
10272           characters (for example, single-byte as opposed to multi-byte characters in  
10273           arguments).10274           **LC\_MESSAGES**10275           Determine the locale that should be used to affect the format and contents of  
10276           diagnostic messages written to standard error.

10277 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

10278 **ASYNCHRONOUS EVENTS**

10279 Default.

10280 **STDOUT**

10281 Not used.

10282 **STDERR**

10283 Used only for diagnostic messages.

10284 **OUTPUT FILES**

10285 None.

10286 **EXTENDED DESCRIPTION**

10287 MAN The *mode* operand shall be either a *symbolic\_mode* expression or a non-negative octal integer. The

10288 *symbolic\_mode* form is described by the grammar later in this section.

### 10289 **Notes to Reviewers**

10290 *This section with side shading will not appear in the final copy. - Ed.*

10291 Please check that the following sentence should have "clause" in a special font.

10292 Each **clause** shall specify an operation to be performed on the current file mode bits of each *file*.

10293 The operations shall be performed on each *file* in the order in which the **clauses** are specified.

10294 The *who* symbols *u*, *g*, and *o* shall specify the *user*, *group*, and *other* parts of the file mode bits,

10295 respectively. A *who* consisting of the symbol *a* shall be equivalent to **ugo**.

10296 The *perm* symbols *r*, *w*, and *x* represent the *read*, *write*, and *execute/search* portions of file mode

10297 bits, respectively. The *perm* symbol *s* shall represent the *set-user-ID-on-execution* (when **who**

10298 contains or implies *u*) and *set-group-ID-on-execution* (when **who** contains or implies *g*) bits.

10299 The **perm** symbol *X* shall represent the execute/search portion of the file mode bits if the file is a

10300 directory or if the current (unmodified) file mode bits have at least one of the execute bits

10301 (*S\_IXUSR*, *S\_IXGRP*, or *S\_IXOTH*) set. It shall be ignored if the file is not a directory and none of

10302 the execute bits are set in the current file mode bits.

10303 The **permcop** symbols *u*, *g*, and *o* shall represent the current permissions associated with the

10304 user, group, and other parts of the file mode bits, respectively. For the remainder of this section,

10305 **perm** refers to the non-terminals **perm** and **permcop** in the grammar.

10306 If multiple **actionlists** are grouped with a single **wholist** in the grammar, each **actionlist** shall be

10307 applied in the order specified with that **wholist**. The *op* symbols shall represent the operation

10308 performed, as follows:

10309 + If **perm** is not specified, the '+' operation shall not change the file mode bits.

10310 If **who** is not specified, the file mode bits represented by **perm** for the owner, group, and

10311 other permissions, except for those with corresponding bits in the file mode creation mask

10312 of the invoking process, shall be set.

10313 Otherwise, the file mode bits represented by the specified **who** and **perm** values shall be set.

10314 – If **perm** is not specified, the '-' operation shall not change the file mode bits.

10315 If **who** is not specified, the file mode bits represented by **perm** for the owner, group, and

10316 other permissions, except for those with corresponding bits in the file mode creation mask

10317 of the invoking process, shall be cleared.

10318

10319

10320

10321

10322

10323

10324

10325

10326

10327

10328

10329

10330

10331

10332

10333

10334

10335

10336

10337

10338

10339

10340

10341

10342

10343

10344

10345

10346

10347

10348

10349

10350

10351

10352

10353

10354

10355

10356

10357

10358

Otherwise, the file mode bits represented by the specified **who** and **perm** values shall be cleared.

= Clear the file mode bits specified by the **who** value, or, if no **who** value is specified, all of the file mode bits specified in this volume of IEEE Std. 1003.1-200x.

If **perm** is not specified, the '=' operation shall make no further modifications to the file mode bits.

If **who** is not specified, the file mode bits represented by **perm** for the owner, group, and other permissions, except for those with corresponding bits in the file mode creation mask of the invoking process, shall be set.

Otherwise, the file mode bits represented by the specified **who** and **perm** values shall be set.

When using the symbolic mode form on a regular file, it is implementation-dependent whether or not:

- Requests to set the set-user-ID-on-execution or set-group-ID-on-execution bit when all execute bits are currently clear and none are being set are ignored.
- Requests to clear all execute bits also clear the set-user-ID-on-execution and set-group-ID-on-execution bits.
- Requests to clear the set-user-ID-on-execution or set-group-ID-on-execution bits when all execute bits are currently clear are ignored. However, if the command *ls -l file* writes an *s* in the position indicating that the set-user-ID-on-execution or set-group-ID-on-execution is set, the commands *chmod u-s file* or *chmod g-s file*, respectively, shall not be ignored.

When using the symbolic mode form on other file types, it is implementation-dependent whether or not requests to set or clear the set-user-ID-on-execution or set-group-ID-on-execution bits are honored.

If the **who** symbol *o* is used in conjunction with the **perm** symbol *s* with no other **who** symbols being specified, the set-user-ID-on-execution and set-group-ID-on-execution bits shall not be modified. It shall not be an error to specify the **who** symbol *o* in conjunction with the **perm** symbol *s*.

For an octal integer *mode* operand, the file mode bits shall be set absolutely.

For each bit set in the octal number, the corresponding file permission bit shown in the following table shall be set; all other file permission bits shall be cleared. For regular files, for each bit set in the octal number corresponding to the set-user-ID-on-execution or the set-group-ID-on-execution, bits shown in the following table shall be set; if these bits are not set in the octal number, they are cleared. For other file types, it is implementation-dependent whether or not requests to set or clear the set-user-ID-on-execution or set-group-ID-on-execution bits are honored.

| Octal | Mode Bit | Octal | Mode Bit | Octal | Mode Bit | Octal | Mode Bit |
|-------|----------|-------|----------|-------|----------|-------|----------|
| 4000  | S_ISUID  | 0400  | S_IRUSR  | 0040  | S_IRGRP  | 0004  | S_IROTH  |
| 2000  | S_ISGID  | 0200  | S_IWUSR  | 0020  | S_IWGRP  | 0002  | S_IWOTH  |
|       |          | 0100  | S_IXUSR  | 0010  | S_IXGRP  | 0001  | S_IXOTH  |

When bits are set in the octal number other than those listed in the table above, the behavior is unspecified.

MAN

10359 **Grammar for chmod**

10360 The grammar and lexical conventions in this section describe the syntax for the *symbolic\_mode*  
 10361 operand. The general conventions for this style of grammar are described in Section 1.10 on page  
 10362 24. A valid *symbolic\_mode* can be represented as the non-terminal symbol *symbolic\_mode* in the  
 10363 grammar. This formal syntax shall take precedence over the preceding text syntax description.

10364 The lexical processing is based entirely on single characters. Implementations need not allow  
 10365 blank characters within the single argument being processed.

```

10366 %start symbolic_mode
10367 %%

10368 symbolic_mode : section
10369 | symbolic_mode ',' section
10370 ;

10371 section : actionlist
10372 | wholist actionlist
10373 ;

10374 wholist : who
10375 | wholist who
10376 ;

10377 who : 'u' | 'g' | 'o' | 'a'
10378 ;

10379 actionlist : action
10380 | actionlist action
10381 ;

10382 action : op
10383 | op permlist
10384 | op permcopy
10385 ;

10386 permcopy : 'u' | 'g' | 'o'
10387 ;

10388 op : '+' | '-' | '='
10389 ;

10390 permlist : perm
10391 | perm permlist
10392 ;

10393 perm : 'r' | 'w' | 'x' | 'X' | 's'
10394 ;

```

10395 **EXIT STATUS**

10396 The following exit values shall be returned:

- 10397     **0** The utility executed successfully and all requested changes were made.
- 10398     **>0** An error occurred.



## 10399 CONSEQUENCES OF ERRORS

10400 If, when invoked with the **-R** option, *chmod* attempts but fails to change the mode of a particular  
 10401 file in a specified file hierarchy, it shall continue to process the remaining files in the hierarchy,  
 10402 affecting the final exit status. If *chmod* cannot read or search a directory within a hierarchy, it  
 10403 shall continue to process the other parts of the hierarchy that are accessible.

## 10404 APPLICATION USAGE

10405 Some implementations of the *chmod* utility change the mode of a directory before the files in the  
 10406 directory when performing a recursive (**-R** option) change; others change the directory mode  
 10407 after the files in the directory. If an application tries to remove read or search permission for a  
 10408 file hierarchy, the removal attempt fails if the directory is changed first; on the other hand, trying  
 10409 to re-enable permissions to a restricted hierarchy fails if directories are changed last. Users  
 10410 should not try to make a hierarchy inaccessible to themselves.

10411 Some implementations of *chmod* never used the process' *umask* when changing modes; systems  
 10412 conformant with this volume of IEEE Std. 1003.1-200x do so when **who** is not specified. Note the  
 10413 difference between:

10414 `chmod a-w file`

10415 which removes all write permissions, and:

10416 `chmod -- -w file`

10417 which removes write permissions that would be allowed if **file** was created with the same  
 10418 *umask*.

10419 Portable applications should never assume that they know how the set-user-ID and set-group-  
 10420 ID bits on directories are interpreted.

## 10421 EXAMPLES

| Mode               | Results                                                                                                  |
|--------------------|----------------------------------------------------------------------------------------------------------|
| <code>a+=</code>   | Equivalent to <code>a+,a=</code> ; clears all file mode bits.                                            |
| <code>go+-w</code> | Equivalent to <code>go+,go-w</code> ; clears group and other write bits.                                 |
| <code>g=o-w</code> | Equivalent to <code>g=o,g-w</code> ; sets group bit to match other bits and then clears group write bit. |
| <code>g-r+w</code> | Equivalent to <code>g-r,g+w</code> ; clears group read bit and sets group write bit.                     |
| <code>=g</code>    | Sets owner bits to match group bits and sets other bits to match group bits.                             |

## 10432 RATIONALE

10433 The functionality of *chmod* is described substantially through references to concepts defined in  
 10434 the System Interfaces volume of IEEE Std. 1003.1-200x. In this way, there is less duplication of  
 10435 effort required for describing the interactions of permissions, and so on. However, the behavior  
 10436 of this utility is not described in terms of the *chmod()* function from the System Interfaces  
 10437 volume of IEEE Std. 1003.1-200x because that specification requires certain side effects upon  
 10438 alternate file access control mechanisms that might not be appropriate, depending on the  
 10439 implementation.

10440 Implementations that support mandatory file and record locking as specified by the 1984  
 10441 /usr/group standard historically used the combination of set-group-ID bit set and group  
 10442 execute bit clear to indicate mandatory locking. This condition is usually set or cleared with the  
 10443 symbolic mode *perm* symbol **l** instead of the *perm* symbols **s** and **x** so that the mandatory locking  
 10444 mode is not changed without explicit indication that that was what the user intended. Therefore,

10445 the details on how the implementation treats these conditions must be defined in the  
 10446 documentation. This volume of IEEE Std. 1003.1-200x does not require mandatory locking (nor  
 10447 does the System Interfaces volume of IEEE Std. 1003.1-200x), but does allow it as an extension.  
 10448 However, this volume of IEEE Std. 1003.1-200x does require that the *ls* and *chmod* utilities work  
 10449 consistently in this area. If *ls -l file* indicates that the set-group-ID bit is set, *chmod g-s file* must  
 10450 clear it (assuming appropriate privileges exist to change modes).

10451 The System V and BSD versions use different exit status codes. Some implementations used the  
 10452 exit status as a count of the number of errors that occurred; this practice is unworkable since it  
 10453 can overflow the range of valid exit status values. This problem is avoided here by specifying  
 10454 only 0 and >0 as exit values.

10455 The System Interfaces volume of IEEE Std. 1003.1-200x indicates that implementation-dependent  
 10456 restrictions may cause the S\_ISUID and S\_ISGID bits to be ignored. This volume of  
 10457 IEEE Std. 1003.1-200x allows the *chmod* utility to choose to modify these bits before calling  
 10458 *chmod()* (or some function providing equivalent capabilities) for non-regular files. Among other  
 10459 things, this allows implementations that use the set-user-ID and set-group-ID bits on directories  
 10460 to enable extended features to handle these extensions in an intelligent manner.

10461 The **X** *perm* symbol was adopted from BSD-based systems because it provides commonly  
 10462 desired functionality when doing recursive (**-R** option) modifications. Similar functionality is  
 10463 not provided by the *find* utility. Historical BSD versions of *chmod*, however, only supported **X**  
 10464 with *op+*; it has been extended in this volume of IEEE Std. 1003.1-200x because it is also useful  
 10465 with *op=*. (It has also been added for *op-* even though it duplicates **x**, in this case, because it is  
 10466 intuitive and easier to explain.)

10467 The grammar was extended with the *permcop*y non-terminal to allow historical-practice forms of  
 10468 symbolic modes like *o=u -g* (that is, set the “other” permissions to the permissions of “owner”  
 10469 minus the permissions of “group”).

#### 10470 FUTURE DIRECTIONS

10471 None.

#### 10472 SEE ALSO

10473 *ls*, *umask*, the System Interfaces volume of IEEE Std. 1003.1-200x, *chmod()*

#### 10474 CHANGE HISTORY

10475 First released in Issue 2.

#### 10476 Issue 4

10477 Aligned with the ISO/IEC 9945-2:1993 standard.

#### 10478 Issue 6

10479 The following new requirements on POSIX implementations derive from alignment with the  
 10480 Single UNIX Specification:

- 10481 • Octal modes have been kept and made mandatory despite being marked obsolescent in the
- 10482 previous version of this volume of IEEE Std. 1003.1-200x.

10483 **NAME**10484 `chown` — change the file ownership10485 **SYNOPSIS**10486 `chown -hR owner[:group] file ...`10487 `chown -R [-H | -L | -P ] owner[:group] file ...`10488 **DESCRIPTION**10489 The *chown* utility shall set the user ID of the file named by each *file* operand to the user ID  
10490 specified by the *owner* operand.10491 For each *file* operand, it shall perform actions equivalent to the *chown()* function defined in the  
10492 System Interfaces volume of IEEE Std. 1003.1-200x, called with the following arguments:

- 10493 1. The *file* operand shall be used as the *path* argument.
- 10494 2. The user ID indicated by the *owner* portion of the first operand shall be used as the *owner*  
10495 argument.
- 10496 3. If the *group* portion of the first operand is given, the group ID indicated by it shall be used  
10497 as the *group* argument; otherwise, the group ID of the file shall be used as the *group*  
10498 argument.

10499 Unless *chown* is invoked by a process with appropriate privileges, the set-user-ID and set-  
10500 group-ID bits of a regular file shall be cleared upon successful completion; the set-user-ID and  
10501 set-group-ID bits of other file types may be cleared.

10502 **OPTIONS**

10503 The *chown* utility shall conform to the System Interface Definitions volume of  
10504 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

10505 The following options shall be supported by the implementation:

- 10506 **-h** If the system supports user IDs for symbolic links, for each *file* operand that names  
10507 a file of type symbolic link, *chown* shall attempt to set the user ID of the symbolic  
10508 link. If the system supports group IDs for symbolic links, and a group ID was  
10509 specified, for each *file* operand that names a file of type symbolic link, *chown* shall  
10510 attempt to set the group ID of the symbolic link. If the system does not support  
10511 user or group IDs for symbolic links, for each *file* operand that names a file of type  
10512 symbolic link, *chown* shall do nothing more with the current file and shall go on to  
10513 any remaining files.
- 10514 **-H** If the **-R** option is specified and a symbolic link referencing a file of type directory  
10515 is specified on the command line, *chown* shall change the user ID (and group ID, if  
10516 specified) of the directory referenced by the symbolic link and all files in the file  
10517 hierarchy below it.
- 10518 **-L** If the **-R** option is specified and a symbolic link referencing a file of type directory  
10519 is specified on the command line or encountered during the traversal of a file  
10520 hierarchy, *chown* shall change the user ID (and group ID, if specified) of the  
10521 directory referenced by the symbolic link and all files in the file hierarchy below it.
- 10522 **-P** If the **-R** option is specified and a symbolic link is specified on the command line  
10523 or encountered during the traversal of a file hierarchy, *chown* shall change the  
10524 owner ID (and group ID, if specified) of the symbolic link if the system supports  
10525 this operation. The *chown* utility shall not follow the symbolic link to any other  
10526 part of the file hierarchy.

10527       **-R**           Recursively change file user and group IDs. For each *file* operand that names a  
 10528                   directory, *chown* shall change the user ID (and group ID, if specified) of the  
 10529                   directory and all files in the file hierarchy below it. Unless a **-H**, **-L**, or **-P** option is  
 10530                   specified, it is unspecified which of these options will be used as the default.

10531           Specifying more than one of the mutually-exclusive options **-H**, **-L**, and **-P** shall not be  
 10532           considered an error. The last option specified shall determine the behavior of the utility.

### 10533 OPERANDS

10534           The following operands shall be supported:

10535       **owner[:group]** A user ID and optional group ID to be assigned to *file*. The application shall  
 10536                   ensure that the *owner* portion of this operand is a user name from the user database  
 10537                   or a numeric user ID. Either specifies a user ID to be given to each file named by  
 10538                   one of the *file* operands. If a numeric *owner* operand exists in the user database as a  
 10539                   user name, the user ID number associated with that user name is used as the user  
 10540                   ID. Similarly, if the *group* portion of this operand is present, it shall be a group  
 10541                   name from the group database or a numeric group ID. Either specifies a group ID  
 10542                   to be given to each file. If a numeric group operand exists in the group database as  
 10543                   a group name, the group ID number associated with that group name shall be used  
 10544                   as the group ID.

10545       **file**           A path name of a file whose user ID is to be modified.

### 10546 STDIN

10547           Not used.

### 10548 INPUT FILES

10549           None.

### 10550 ENVIRONMENT VARIABLES

10551           The following environment variables shall affect the execution of *chown*:

10552       **LANG**           Provide a default value for the internationalization variables that are unset or null.  
 10553                   If *LANG* is unset or null, the corresponding value from the implementation-  
 10554                   dependent default locale shall be used. If any of the internationalization variables  
 10555                   contains an invalid setting, the utility shall behave as if none of the variables had  
 10556                   been defined.

10557       **LC\_ALL**          If set to a non-empty string value, override the values of all the other  
 10558                   internationalization variables.

10559       **LC\_CTYPE**       Determine the locale for the interpretation of sequences of bytes of text data as  
 10560                   characters (for example, single-byte as opposed to multi-byte characters in  
 10561                   arguments).

10562       **LC\_MESSAGES**

10563                   Determine the locale that should be used to affect the format and contents of  
 10564                   diagnostic messages written to standard error.

10565 XSI       **NLSPATH**   Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

### 10566 ASYNCHRONOUS EVENTS

10567           Default.

### 10568 STDOUT

10569           Not used.

10570 **STDERR**

10571           Used only for diagnostic messages.

10572 **OUTPUT FILES**

10573           None.

10574 **EXTENDED DESCRIPTION**

10575           None.

10576 **EXIT STATUS**

10577           The following exit values shall be returned:

10578           0   The utility executed successfully and all requested changes were made.

10579           >0   An error occurred.

10580 **CONSEQUENCES OF ERRORS**

10581           If, when invoked with the **-R** option, *chown* attempts but fails to change the user ID or, if the *group* operand is specified, group ID, of a particular file in a specified file hierarchy, it shall continue to process the remaining files in the hierarchy.

10584           If *chown* cannot read or search a directory within a hierarchy, it shall continue to process the other parts of the hierarchy that are accessible.

10586 **APPLICATION USAGE**

10587           Only the owner of a file or the user with appropriate privileges may change the owner or group of a file.

10589           Some systems restrict the use of *chown* to a user with appropriate privileges.

10590 **EXAMPLES**

10591           None.

10592 **RATIONALE**

10593           The System V and BSD versions use different exit status codes. Some implementations used the exit status as a count of the number of errors that occurred; this practice is unworkable since it can overflow the range of valid exit status values. These are masked by specifying only 0 and >0 as exit values.

10597           The functionality of *chown* is described substantially through references to functions in the System Interfaces volume of IEEE Std. 1003.1-200x. In this way, there is no duplication of effort required for describing the interactions of permissions, multiple groups, and so on.

10600           The 4.3 BSD method of specifying both owner and group was included in this volume of IEEE Std. 1003.1-200x because:

10602           • There are cases where the desired end condition could not be achieved using the *chgrp* and *chown* (that only changed the user ID) utilities. (If the current owner is not a member of the desired group and the desired owner is not a member of the current group, the *chown()* function could fail unless both owner and group are changed at the same time.)

10606           • Even if they could be changed independently, in cases where both are being changed, there is a 100% performance penalty caused by being forced to invoke both utilities.

10608           The BSD syntax *user[.group]* was changed to *user[:group]* in this volume of IEEE Std. 1003.1-200x because the period is a valid character in login names (as specified by the System Interface Definitions volume of IEEE Std. 1003.1-200x, login names consist of characters in the portable file name character set). The colon character was chosen as the replacement for the period character because it would never be allowed as a character in a user name or group name on historical implementations.

10614 The **-R** option is considered by some observers as an undesirable departure from the historical  
10615 UNIX system tools approach; since a tool, *find*, already exists to recurse over directories, there  
10616 seemed to be no good reason to require other tools to have to duplicate that functionality.  
10617 However, the **-R** option was deemed an important user convenience, is far more efficient than  
10618 forking a separate process for each element of the directory hierarchy, and is in widespread  
10619 historical use.

10620 **FUTURE DIRECTIONS**

10621 None.

10622 **SEE ALSO**

10623 *chmod*, *chgrp*, the System Interfaces volume of IEEE Std. 1003.1-200x, *chown()*

10624 **CHANGE HISTORY**

10625 First released in Issue 2.

10626 **Issue 4**

10627 Aligned with the ISO/IEC 9945-2: 1993 standard.

10628 **Issue 6**

10629 New options **-h**, **-H**, **-L**, and **-P** are added to align with the IEEE P1003.2b draft standard. These  
10630 options affect the processing of symbolic links.

10631 The normative text is reworded to avoid use of the term “must” for application requirements.

10632 **NAME**

10633 cksum — write file checksums and sizes

10634 **SYNOPSIS**10635 cksum [*file* ...]10636 **DESCRIPTION**

10637 The *cksum* utility shall calculate and write to standard output a cyclic redundancy check (CRC)  
 10638 for each input file, and also write to standard output the number of octets in each file. The CRC  
 10639 used is based on the polynomial used for CRC error checking in the ISO/IEC 8802-3:1996  
 10640 standard (Ethernet).

10641 The encoding for the CRC checksum is defined by the generating polynomial:

10642  $G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

10643 Mathematically, the CRC value corresponding to a given file shall be defined by the following  
 10644 procedure:

- 10645 1. The *n* bits to be evaluated are considered to be the coefficients of a mod 2 polynomial *M*(*x*)  
 10646 of degree *n*−1. These *n* bits are the bits from the file, with the most significant bit being the  
 10647 most significant bit of the first octet of the file and the last bit being the least significant bit  
 10648 of the last octet, padded with zero bits (if necessary) to achieve an integral number of  
 10649 octets, followed by one or more octets representing the length of the file as a binary value,  
 10650 least significant octet first. The smallest number of octets capable of representing this  
 10651 integer shall be used.
- 10652 2. *M*(*x*) is multiplied by  $x^{32}$  (that is, shifted left 32 bits) and divided by *G*(*x*) using mod 2  
 10653 division, producing a remainder *R*(*x*) of degree ≤ 31.
- 10654 3. The coefficients of *R*(*x*) are considered to be a 32-bit sequence.
- 10655 4. The bit sequence is complemented and the result is the CRC.

10656 **OPTIONS**

10657 None.

10658 **OPERANDS**

10659 The following operand shall be supported:

10660 *file* A path name of a file to be checked. If no *file* operands are specified, the standard  
 10661 input is used.

10662 **STDIN**10663 The standard input is used only if no *file* operands are specified. See the INPUT FILES section.10664 **INPUT FILES**

10665 The input files can be any file type.

10666 **ENVIRONMENT VARIABLES**10667 The following environment variables shall affect the execution of *cksum*:

10668 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 10669 If *LANG* is unset or null, the corresponding value from the implementation-  
 10670 dependent default locale shall be used. If any of the internationalization variables  
 10671 contains an invalid setting, the utility shall behave as if none of the variables had  
 10672 been defined.

10673 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 10674 internationalization variables.

10675        *LC\_CTYPE*   Determine the locale for the interpretation of sequences of bytes of text data as  
 10676                   characters (for example, single-byte as opposed to multi-byte characters in  
 10677                   arguments).

10678        *LC\_MESSAGES*  
 10679                   Determine the locale that should be used to affect the format and contents of  
 10680                   diagnostic messages written to standard error.

10681 XSI        *NLSPATH*   Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

10682 **ASYNCHRONOUS EVENTS**  
 10683        Default.

10684 **STDOUT**  
 10685        For each file processed successfully, the *cksum* utility shall write in the following format:  
 10686        "%u %d %s\n", <checksum>, <# of octets>, <pathname>  
 10687        If no *file* operand was specified, the path name and its leading <space> shall be omitted.

10688 **STDERR**  
 10689        Used only for diagnostic messages.

10690 **OUTPUT FILES**  
 10691        None.

10692 **EXTENDED DESCRIPTION**  
 10693        None.

10694 **EXIT STATUS**  
 10695        The following exit values shall be returned:  
 10696        0   All files were processed successfully.  
 10697        >0   An error occurred.

10698 **CONSEQUENCES OF ERRORS**  
 10699        Default.

10700 **APPLICATION USAGE**  
 10701        The *cksum* utility is typically used to quickly compare a suspect file against a trusted version of  
 10702        the same, such as to ensure that files transmitted over noisy media arrive intact. However, this  
 10703        comparison cannot be considered cryptographically secure. The chances of a damaged file  
 10704        producing the same CRC as the original are small; deliberate deception is difficult, but probably  
 10705        not impossible.

10706        Although input files to *cksum* can be any type, the results need not be what would be expected  
 10707        on character special device files or on file types not described by the System Interfaces volume of  
 10708        IEEE Std. 1003.1-200x. Since this volume of IEEE Std. 1003.1-200x does not specify the block size  
 10709        used when doing input, checksums of character special files need not process all of the data in  
 10710        those files.

10711        The algorithm is expressed in terms of a bitstream divided into octets. If a file is transmitted  
 10712        between two systems and undergoes any data transformation (such as moving 8-bit characters  
 10713        into 9-bit bytes or changing little-endian byte ordering to big-endian), identical CRC values  
 10714        cannot be expected. Implementations performing such transformations may extend *cksum* to  
 10715        handle such situations.



10716 **EXAMPLES**

10717       None.

10718 **RATIONALE**

10719       The following C-language program can be used as a model to describe the algorithm. It assumes  
 10720       that a **char** is one octet. It also assumes that the entire file is available for one pass through the  
 10721       function. This was done for simplicity in demonstrating the algorithm, rather than as an  
 10722       implementation model.

```

10723 static unsigned long crctab[] = {
10724 0x00000000,
10725 0x04c11db7, 0x09823b6e, 0x0d4326d9, 0x130476dc, 0x17c56b6b,
10726 0x1a864db2, 0x1e475005, 0x2608edb8, 0x22c9f00f, 0x2f8ad6d6,
10727 0x2b4bcb61, 0x350c9b64, 0x31cd86d3, 0x3c8ea00a, 0x384fbd8d,
10728 0x4c11db70, 0x48d0c6c7, 0x4593e01e, 0x4152fda9, 0x5f15adac,
10729 0x5bd4b01b, 0x569796c2, 0x52568b75, 0x6a1936c8, 0x6ed82b7f,
10730 0x639b0da6, 0x675a1011, 0x791d4014, 0x7ddc5da3, 0x709f7b7a,
10731 0x745e66cd, 0x9823b6e0, 0x9ce2ab57, 0x91a18d8e, 0x95609039,
10732 0x8b27c03c, 0x8fe6dd8b, 0x82a5fb52, 0x8664e6e5, 0xbe2b5b58,
10733 0xbaea46ef, 0xb7a96036, 0xb3687d81, 0xad2f2d84, 0xa9ee3033,
10734 0xa4ad16ea, 0xa06c0b5d, 0xd4326d90, 0xd0f37027, 0xddb056fe,
10735 0xd9714b49, 0xc7361b4c, 0xc3f706fb, 0xceb42022, 0xca753d95,
10736 0xf23a8028, 0xf6fb9d9f, 0xfbb8bb46, 0xff79a6f1, 0xe13ef6f4,
10737 0xe5ffe43, 0xe8bccd9a, 0xec7dd02d, 0x34867077, 0x30476dc0,
10738 0x3d044b19, 0x39c556ae, 0x278206ab, 0x23431b1c, 0x2e003dc5,
10739 0x2ac12072, 0x128e9dcf, 0x164f8078, 0x1b0ca6a1, 0x1fcd8bb16,
10740 0x018aeb13, 0x054bf6a4, 0x0808d07d, 0x0cc9cdca, 0x7897ab07,
10741 0x7c56b6b0, 0x71159069, 0x75d48dde, 0x6b93dddb, 0x6f52c06c,
10742 0x6211e6b5, 0x66d0fb02, 0x5e9f46bf, 0x5a5e5b08, 0x571d7dd1,
10743 0x53dc6066, 0x4d9b3063, 0x495a2dd4, 0x44190b0d, 0x40d816ba,
10744 0xaca5c697, 0xa864db20, 0xa527fdf9, 0xa1e6e04e, 0xbfa1b04b,
10745 0xbb60adfc, 0xb6238b25, 0xb2e29692, 0x8aad2b2f, 0x8e6c3698,
10746 0x832f1041, 0x87ee0df6, 0x99a95df3, 0x9d684044, 0x902b669d,
10747 0x94ea7b2a, 0xe0b41de7, 0xe4750050, 0xe9362689, 0xedf73b3e,
10748 0xf3b06b3b, 0xf771768c, 0xfa325055, 0xfef34de2, 0xc6bcf05f,
10749 0xc27dede8, 0xcf3ecb31, 0xcbffd686, 0xd5b88683, 0xd1799b34,
10750 0xdc3abded, 0xd8fba05a, 0x690ce0ee, 0x6dcd5f59, 0x608edb80,
10751 0x644fc637, 0x7a089632, 0x7ec98b85, 0x738aad5c, 0x774bb0eb,
10752 0x4f040d56, 0x4bc510e1, 0x46863638, 0x42472b8f, 0x5c007b8a,
10753 0x58c1663d, 0x558240e4, 0x51435d53, 0x251d3b9e, 0x21dc2629,
10754 0x2c9f00f0, 0x285e1d47, 0x36194d42, 0x32d850f5, 0x3f9b762c,
10755 0x3b5a6b9b, 0x0315d626, 0x07d4cb91, 0x0a97ed48, 0x0e56f0ff,
10756 0x1011a0fa, 0x14d0bd4d, 0x19939b94, 0x1d528623, 0xf12f560e,
10757 0xf5ee4bb9, 0xf8ad6d60, 0xfc6c70d7, 0xe22b20d2, 0xe6ea3d65,
10758 0xeba91bbc, 0xef68060b, 0xd727bbb6, 0xd3e6a601, 0xdea580d8,
10759 0xda649d6f, 0xc423cd6a, 0xc0e2d0dd, 0xcda1f604, 0xc960ebb3,
10760 0xbd3e8d7e, 0xb9ff90c9, 0xb4bcb610, 0xb07daba7, 0xae3afba2,
10761 0xaafbe615, 0xa7b8c0cc, 0xa379dd7b, 0x9b3660c6, 0x9ff77d71,
10762 0x92b45ba8, 0x9675461f, 0x8832161a, 0x8cf30bad, 0x81b02d74,
10763 0x857130c3, 0x5d8a9099, 0x594b8d2e, 0x5408abf7, 0x50c9b640,
10764 0x4e8ee645, 0x4a4ffbf2, 0x470cdd2b, 0x43cdc09c, 0x7b827d21,
10765 0x7f436096, 0x7200464f, 0x76c15bf8, 0x68860bfd, 0x6c47164a,
10766 0x61043093, 0x65c52d24, 0x119b4be9, 0x155a565e, 0x18197087,

```

```

10767 0x1cd86d30, 0x029f3d35, 0x065e2082, 0x0b1d065b, 0x0fdc1bec,
10768 0x3793a651, 0x3352bbe6, 0x3e119d3f, 0x3ad08088, 0x2497d08d,
10769 0x2056cd3a, 0x2d15ebe3, 0x29d4f654, 0xc5a92679, 0xc1683bce,
10770 0xcc2b1d17, 0xc8ea00a0, 0xd6ad50a5, 0xd26c4d12, 0xdf2f6bcb,
10771 0xdbee767c, 0xe3alc1c1, 0xe760d676, 0xea23f0af, 0xee2ed18,
10772 0xf0a5bd1d, 0xf464a0aa, 0xf9278673, 0xfde69bc4, 0x89b8fd09,
10773 0x8d79e0be, 0x803ac667, 0x84fbbdbd, 0x9abc8bd5, 0x9e7d9662,
10774 0x933eb0bb, 0x97ffad0c, 0xafb010b1, 0xab710d06, 0xa6322bdf,
10775 0xa2f33668, 0xbcb4666d, 0xb8757bda, 0xb5365d03, 0xb1f740b4
10776 };

10777 unsigned long memcrc(const unsigned char *b, size_t n)
10778 {
10779 /* Input arguments:
10780 * const char* b == byte sequence to checksum
10781 * size_t n == length of sequence
10782 */

10783 register unsigned int i, c, s = 0;

10784 for (i = n; i > 0; --i) {
10785 c = (unsigned int)(*b++);
10786 s = (s << 8) ^ crctab[(s >> 24) ^ c];
10787 }

10788 /* Extend with the length of the string. */
10789 while (n != 0) {
10790 c = n & 0377;
10791 n >>= 8;
10792 s = (s << 8) ^ crctab[(s >> 24) ^ c];
10793 }

10794 return ~s;
10795 }

```

The historical practice of writing the number of “blocks” has been changed to writing the number of octets, since the latter is not only more useful, but also since historical implementations have not been consistent in defining what a “block” meant. Octets are used instead of bytes because bytes can differ in size between systems.

The algorithm used was selected to increase the operational robustness of *cksum*. Neither the System V nor BSD *sum* algorithm was selected. Since each of these was different and each was the default behavior on those systems, no realistic compromise was available if either were selected—some set of historical applications would break. Therefore, the name was changed to *cksum*. Although the historical *sum* commands will probably continue to be provided for many years, programs designed for portability across systems should use the new name.

The algorithm selected is based on that used by the ISO/IEC 8802-3: 1996 standard (Ethernet) for the frame check sequence field. The algorithm used does not match the technical definition of a *checksum*; the term is used for historical reasons. The length of the file is included in the CRC calculation because this parallels inclusion of a length field by Ethernet in its CRC, but also because it guards against inadvertent collisions between files that begin with different series of zero octets. The chance that two different files produce identical CRCs is much greater when their lengths are not considered. Keeping the length and the checksum of the file itself separate would yield a slightly more robust algorithm, but historical usage has always been that a single number (the checksum as printed) represents the signature of the file. It was decided that

```

10815 historical usage was the more important consideration.
10816 Early proposals contained modifications to the Ethernet algorithm that involved extracting table
10817 values whenever an intermediate result became zero. This was demonstrated to be less robust
10818 than the current method and mathematically difficult to describe or justify.
10819 The calculation used is identical to that given in pseudo-code in the referenced Sarwate Article.
10820 The pseudo-code rendition is:
10821 X <- 0; Y <- 0;
10822 for i <- m -1 step -1 until 0 do
10823 begin
10824 T <- X(1) ^ A[i];
10825 X(1) <- X(0); X(0) <- Y(1); Y(1) <- Y(0); Y(0) <- 0;
10826 comment: f[T] and f'[T] denote the T-th words in the
10827 table f and f' ;
10828 X <- X ^ f[T]; Y <- Y ^ f'[T];
10829 end
10830 The pseudo-code is reproduced exactly as given; however, note that in the case of cksum, A[i]
10831 represents a byte of the file, the words X and Y are treated as a single 32-bit value, and the tables
10832 f and f' are a single table containing 32-bit values.
10833 The referenced Sarwate Article also discusses generating the table.
10834 FUTURE DIRECTIONS
10835 None.
10836 SEE ALSO
10837 None.
10838 CHANGE HISTORY
10839 First released in Issue 4.

```

10840 **NAME**

10841 cmp — compare two files

10842 **SYNOPSIS**10843 cmp [ -l | -s ] *file1 file2*10844 **DESCRIPTION**

10845 The *cmp* utility shall compare two files. The *cmp* utility writes no output if the files are the same.  
 10846 Under default options, if they differ, it shall write to standard output the byte and line number at  
 10847 which the first difference occurred. Bytes and lines shall be numbered beginning with 1.

10848 **OPTIONS**

10849 The *cmp* utility shall conform to the System Interface Definitions volume of  
 10850 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

10851 The following options shall be supported:

10852 **-l** (Lowercase ell.) Write the byte number (decimal) and the differing bytes (octal) for  
 10853 each difference.  
 10854 **-s** Write nothing for differing files; return exit status only.

10855 **OPERANDS**

10856 The following operands shall be supported:

10857 ***file1*** A path name of the first file to be compared. If *file1* is '-', the standard input shall  
 10858 be used.  
 10859 ***file2*** A path name of the second file to be compared. If *file2* is '-', the standard input  
 10860 shall be used.

10861 If both *file1* and *file2* refer to standard input or refer to the same FIFO special, block special, or  
 10862 character special file, the results are undefined.

10863 **STDIN**

10864 The standard input shall be used only if the *file1* or *file2* operand refers to standard input. See the  
 10865 INPUT FILES section.

10866 **INPUT FILES**

10867 The input files can be any file type.

10868 **ENVIRONMENT VARIABLES**10869 The following environment variables shall affect the execution of *cmp*:

10870 ***LANG*** Provide a default value for the internationalization variables that are unset or null.  
 10871 If *LANG* is unset or null, the corresponding value from the implementation-  
 10872 dependent default locale shall be used. If any of the internationalization variables  
 10873 contains an invalid setting, the utility shall behave as if none of the variables had  
 10874 been defined.

10875 ***LC\_ALL*** If set to a non-empty string value, override the values of all the other  
 10876 internationalization variables.

10877 ***LC\_CTYPE*** Determine the locale for the interpretation of sequences of bytes of text data as  
 10878 characters (for example, single-byte as opposed to multi-byte characters in  
 10879 arguments).

10880 ***LC\_MESSAGES***

10881 Determine the locale that should be used to affect the format and contents of  
 10882 diagnostic messages written to standard error and informative messages written to  
 10883 standard output.

10884 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

10885 **ASYNCHRONOUS EVENTS**

10886 Default.

10887 **STDOUT**

10888 In the POSIX locale, results of the comparison shall be written to standard output. When no  
 10889 options are used, the format shall be:

10890 "%s %s differ: char %d, line %d\n", *file1*, *file2*,  
 10891 <byte number>, <line number>

10892 When the **-l** option is used, the format shall be:

10893 "%d %o %o\n", <byte number>, <differing byte>,  
 10894 <differing byte>

10895 for each byte that differs. The first <differing byte> number is from *file1* while the second is from  
 10896 *file2*. In both cases, <byte number> shall be relative to the beginning of the file, beginning with 1.

10897 No output shall be written to standard output when the **-s** option is used.

10898 **STDERR**

10899 Used only for diagnostic messages. If *file1* and *file2* are identical for the entire length of the  
 10900 shorter file, in the POSIX locale the following diagnostic message shall be written, unless the **-s**  
 10901 option is specified:

10902 "cmp: EOF on %s%s\n", <name of shorter file>, <additional info>

10903 The <additional info> field shall either be null or a string that starts with a <blank> character and  
 10904 contains no <newline> characters. Some systems report on the number of lines in this case.

10905 **OUTPUT FILES**

10906 None.

10907 **EXTENDED DESCRIPTION**

10908 None.

10909 **EXIT STATUS**

10910 The following exit values shall be returned:

10911 0 The files are identical.

10912 1 The files are different; this includes the case where one file is identical to the first part of the  
 10913 other.

10914 >1 An error occurred.

10915 **CONSEQUENCES OF ERRORS**

10916 Default.

10917 **APPLICATION USAGE**

10918 Although input files to *cmp* can be any type, the results might not be what would be expected on  
 10919 character special device files or on file types not described by the System Interfaces volume of  
 10920 IEEE Std. 1003.1-200x. Since this volume of IEEE Std. 1003.1-200x does not specify the block size  
 10921 used when doing input, comparisons of character special files need not compare all of the data  
 10922 in those files.

10923 For files which are not text files, line numbers simply reflect the presence of a <newline>  
 10924 character, without any implication that the file is organized into lines.

10925 **EXAMPLES**

10926           None.

10927 **RATIONALE**

10928           The global language in Section 1.11 on page 25 indicates that using two mutually-exclusive  
10929           options together produces unspecified results. Some System V implementations consider the  
10930           option usage:

10931           cmp -l -s ...

10932           to be an error. They also treat:

10933           cmp -s -l ...

10934           as if no options were specified. Both of these behaviors are considered bugs, but are allowed.

10935           The word **char** in the standard output format comes from historical usage, even though it is  
10936           actually a byte number. When *cmp* is supported in other locales, implementations are  
10937           encouraged to use the word *byte* or its equivalent in another language. Users should not  
10938           interpret this difference to indicate that the functionality of the utility changed between locales.

10939           Some systems report on the number of lines in the identical-but-shorter file case. This is allowed  
10940           by the inclusion of the <additional info> fields in the output format. The restriction on having a  
10941           leading <blank> and no <newline>s is to make parsing for the file name easier. It is recognized  
10942           that some file names containing white-space characters make parsing difficult anyway, but the  
10943           restriction does aid programs used on systems where the names are predominantly well  
10944           behaved.

10945 **FUTURE DIRECTIONS**

10946           None.

10947 **SEE ALSO**10948           *comm*, *diff*10949 **CHANGE HISTORY**

10950           First released in Issue 2.

10951 **Issue 4**

10952           Aligned with the ISO/IEC 9945-2:1993 standard.

10953 **NAME**

10954           comm — select or reject lines common to two files

10955 **SYNOPSIS**10956           comm [-123] *file1 file2*10957 **DESCRIPTION**

10958           The *comm* utility shall read *file1* and *file2*, which should be ordered in the current collating  
 10959           sequence, and produce three text columns as output: lines only in *file1*, lines only in *file2*, and  
 10960           lines in both files.

10961           If the lines in both files are not ordered according to the collating sequence of the current locale,  
 10962           the results are unspecified.

10963 **OPTIONS**

10964           The *comm* utility shall conform to the System Interface Definitions volume of  
 10965           IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

10966           The following options shall be supported:

10967           -1           Suppress the output column of lines unique to *file1*.

10968           -2           Suppress the output column of lines unique to *file2*.

10969           -3           Suppress the output column of lines duplicated in *file1* and *file2*.

10970 **OPERANDS**

10971           The following operands shall be supported:

10972           *file1*       A path name of the first file to be compared. If *file1* is '-', the standard input is  
 10973           used.

10974           *file2*       A path name of the second file to be compared. If *file2* is '-', the standard input is  
 10975           used.

10976           If both *file1* and *file2* refer to standard input or to the same FIFO special, block special, or  
 10977           character special file, the results are undefined.

10978 **STDIN**

10979           The standard input shall be used only if one of the *file1* or *file2* operands refers to standard input.  
 10980           See the INPUT FILES section.

10981 **INPUT FILES**

10982           The input files shall be text files.

10983 **ENVIRONMENT VARIABLES**

10984           The following environment variables shall affect the execution of *comm*:

10985           *LANG*       Provide a default value for the internationalization variables that are unset or null.  
 10986           If *LANG* is unset or null, the corresponding value from the implementation-  
 10987           dependent default locale shall be used. If any of the internationalization variables  
 10988           contains an invalid setting, the utility shall behave as if none of the variables had  
 10989           been defined.

10990           *LC\_ALL*      If set to a non-empty string value, override the values of all the other  
 10991           internationalization variables.

10992           *LC\_COLLATE*

10993           Determine the locale for the collating sequence *comm* expects to have been used  
 10994           when the input files were sorted.

- 10995 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 10996 characters (for example, single-byte as opposed to multi-byte characters in  
 10997 arguments and input files).
- 10998 **LC\_MESSAGES**  
 10999 Determine the locale that should be used to affect the format and contents of  
 11000 diagnostic messages written to standard error.
- 11001 XSI **NLSPATH** Determine the location of message catalogs for the processing of **LC\_MESSAGES**.
- 11002 **ASYNCHRONOUS EVENTS**
- 11003 Default.
- 11004 **STDOUT**
- 11005 The *comm* utility shall produce output depending on the options selected. If the **-1**, **-2**, and **-3**  
 11006 options are all selected, *comm* shall write nothing to standard output.
- 11007 If the **-1** option is not selected, lines contained only in *file1* shall be written using the format:  
 11008 "%s\n", <line in file1>
- 11009 If the **-2** option is not selected, lines contained only in *file2* are written using the format:  
 11010 "%s%s\n", <lead>, <line in file2>
- 11011 where the string <lead> is as follows:
- 11012 <tab> The **-1** option is not selected.
- 11013 null string The **-1** option is selected.
- 11014 If the **-3** option is not selected, lines contained in both files shall be written using the format:  
 11015 "%s%s\n", <lead>, <line in both>
- 11016 where the string <lead> is as follows:
- 11017 <tab><tab> Neither the **-1** nor the **-2** option is selected.
- 11018 <tab> Exactly one of the **-1** and **-2** options is selected.
- 11019 null string Both the **-1** and **-2** options are selected.
- 11020 If the input files were ordered according to the collating sequence of the current locale, the lines  
 11021 written shall be in the collating sequence of the original lines.
- 11022 **STDERR**
- 11023 Used only for diagnostic messages.
- 11024 **OUTPUT FILES**
- 11025 None.
- 11026 **EXTENDED DESCRIPTION**
- 11027 None.
- 11028 **EXIT STATUS**
- 11029 The following exit values shall be returned:
- 11030 0 All input files were successfully output as specified.
- 11031 >0 An error occurred.



11032 **CONSEQUENCES OF ERRORS**

11033 Default.

11034 **APPLICATION USAGE**11035 If the input files are not properly presorted, the output of *comm* might not be useful.11036 **EXAMPLES**

11037 If a file named **xcu** contains a sorted list of the utilities in this volume of IEEE Std. 1003.1-200x, a  
 11038 file named **xpg3** contains a sorted list of the utilities specified in the X/Open Portability Guide,  
 11039 Issue 3, and a file named **svid89** contains a sorted list of the utilities in the System V Interface  
 11040 Definition Third Edition:

11041 `comm -23 xcu xpg3 | comm -23 - svid89`

11042 would print a list of utilities in this volume of IEEE Std. 1003.1-200x not specified by either of the  
 11043 other documents:

11044 `comm -12 xcu xpg3 | comm -12 - svid89`

11045 would print a list of utilities specified by all three documents, and:

11046 `comm -12 xpg3 svid89 | comm -23 - xcu`

11047 would print a list of utilities specified by both XPG3 and the SVID, but not specified in this  
 11048 volume of IEEE Std. 1003.1-200x.

11049 **RATIONALE**

11050 None.

11051 **FUTURE DIRECTIONS**

11052 None.

11053 **SEE ALSO**11054 *cmp, diff, sort, uniq*11055 **CHANGE HISTORY**

11056 First released in Issue 2.

11057 **Issue 4**

11058 Aligned with the ISO/IEC 9945-2:1993 standard. |

11059 **Issue 6**

11060 The normative text is reworded to avoid use of the term “must” for application requirements. |

11061 **NAME**

11062       command — execute a simple command

11063 **SYNOPSIS**11064       command [-p] *command\_name* [*argument* ...]11065 UP      command [ -v | -V ] *command\_name*

11066

11067 **DESCRIPTION**11068       The *command* utility shall cause the shell to treat the arguments as a simple command, suppressing the shell function lookup that is described in Section 2.9.1.1 on page 69, item 1b.11069  
11070       If the *command\_name* is the same as the name of one of the special built-in utilities, the special  
11071       properties in the enumerated list at the beginning of Section 2.14 on page 96 shall not occur. In  
11072       every other respect, if *command\_name* is not the name of a function, the effect of *command* shall be  
11073       the same as omitting *command*.11074       On systems supporting the User Portability Utilities option, the *command* utility also shall  
11075       provide information concerning how a command name is interpreted by the shell; see -v and  
11076       -V.11077 **OPTIONS**11078       The *command* utility shall conform to the System Interface Definitions volume of  
11079       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

11080       The following options shall be supported:

11081       -p       Perform the command search using a default value for *PATH* that is guaranteed to  
11082       find all of the standard utilities.11083       -v       (On systems supporting the User Portability Utilities option.) Write a string to  
11084       standard output that indicates the path name or command that will be used by the  
11085       shell, in the current shell execution environment (see Section 2.12 on page 90), to  
11086       invoke *command\_name*.11087               • Utilities, regular built-in utilities, *command\_names* including a slash character,  
11088               and any implementation-dependent functions that are found using the *PATH*  
11089               variable (as described in Section 2.9.1.1 on page 69), shall be written as absolute  
11090               path names.11091               • Shell functions, special built-in utilities, regular built-in utilities not associated  
11092               with a *PATH* search, and shell reserved words shall be written as just their  
11093               names.

11094               • An alias shall be written as a command line that represents its alias definition.

11095               • Otherwise, no output shall be written and the exit status shall reflect that the  
11096               name was not found.11097       -V       (On systems supporting the User Portability Utilities option.) Write a string to  
11098       standard output that indicates how the name given in the *command\_name* operand  
11099       will be interpreted by the shell, in the current shell execution environment (see  
11100       Section 2.12 on page 90). Although the format of this string is unspecified, it shall  
11101       indicate in which of the following categories *command\_name* falls and shall include  
11102       the information stated:11103               • Utilities, regular built-in utilities, and any implementation-dependent functions  
11104               that are found using the *PATH* variable (as described in Section 2.9.1.1 on page  
11105               69), shall be identified as such and include the absolute path name in the string.

- 11106 • Other shell functions shall be identified as functions.
- 11107 • Aliases shall be identified as aliases and their definitions included in the string.
- 11108 • Special built-in utilities shall be identified as special built-in utilities.
- 11109 • Regular built-in utilities not associated with a *PATH* search shall be identified
- 11110 as regular built-in utilities. (The term “regular” need not be used.)
- 11111 • Shell reserved words shall be identified as reserved words.

**11112 OPERANDS**

11113 The following operands shall be supported:

11114 *argument* One of the strings treated as an argument to *command\_name*.

11115 *command\_name*

11116 The name of a utility or a special built-in utility.

**11117 STDIN**

11118 Not used.

**11119 INPUT FILES**

11120 None.

**11121 ENVIRONMENT VARIABLES**

11122 The following environment variables shall affect the execution of *command*:

11123 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 11124 If *LANG* is unset or null, the corresponding value from the implementation-  
 11125 dependent default locale shall be used. If any of the internationalization variables  
 11126 contains an invalid setting, the utility shall behave as if none of the variables had  
 11127 been defined.

11128 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 11129 internationalization variables.

11130 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 11131 characters (for example, single-byte as opposed to multi-byte characters in  
 11132 arguments).

11133 *LC\_MESSAGES*

11134 Determine the locale that should be used to affect the format and contents of  
 11135 diagnostic messages written to standard error and informative messages written to  
 11136 standard output.

11137 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

11138 *PATH* Determine the search path used during the command search described in Section  
 11139 2.9.1.1 on page 69, except as described under the *-p* option.

**11140 ASYNCHRONOUS EVENTS**

11141 Default.

**11142 STDOUT**

11143 When the *-v* option is specified, standard output shall be formatted as:

11144 "%s\n", <pathname or command>

11145 When the *-V* option is specified, standard output shall be formatted as:

11146 "%s\n", <unspecified>

11147 **STDERR**

11148       Used only for diagnostic messages.

11149 **OUTPUT FILES**

11150       None.

11151 **EXTENDED DESCRIPTION**

11152       None.

11153 **EXIT STATUS**11154       When the **-v** or **-V** options are specified, the following exit values shall be returned:

11155       0   Successful completion.

11156       >0   The *command\_name* could not be found or an error occurred.

11157       Otherwise, the following exit values shall be returned:

11158       126   The utility specified by *command\_name* was found but could not be invoked.11159       127   An error occurred in the *command* utility or the utility specified by *command\_name* could not  
11160       be found.11161       Otherwise, the exit status of *command* shall be that of the simple command specified by the  
11162       arguments to *command*.11163 **CONSEQUENCES OF ERRORS**

11164       Default.

11165 **APPLICATION USAGE**11166       The order for command search allows functions to override regular built-ins and path searches.  
11167       This utility is necessary to allow functions that have the same name as a utility to call the utility  
11168       (instead of a recursive call to the function).11169       The system default path is available using *getconf*; however, since *getconf* may need to have the  
11170       *PATH* set up before it can be called itself, the following can be used:11171       `command -p getconf _CS_PATH`11172       There are some advantages to suppressing the special characteristics of special built-ins on  
11173       occasion. For example:11174       `command exec > unwritable-file`11175       does not cause a non-interactive script to abort, so that the output status can be checked by the  
11176       script.11177       The *command*, *env*, *nohup*, *time*, and *xargs* utilities have been specified to use exit code 127 if an  
11178       error occurs so that applications can distinguish “failure to find a utility” from “invoked utility  
11179       exited with an error indication”. The value 127 was chosen because it is not commonly used for  
11180       other meanings; most utilities use small values for “normal error conditions” and the values  
11181       above 128 can be confused with termination due to receipt of a signal. The value 126 was chosen  
11182       in a similar manner to indicate that the utility could be found, but not invoked. Some scripts  
11183       produce meaningful error messages differentiating the 126 and 127 cases. The distinction  
11184       between exit codes 126 and 127 is based on KornShell practice that uses 127 when all attempts to  
11185       *exec* the utility fail with [ENOENT], and uses 126 when any attempt to *exec* the utility fails for  
11186       any other reason.11187       Since the **-v** and **-V** options of *command* produce output in relation to the current shell execution  
11188       environment, *command* is generally provided as a shell regular built-in. If it is called in a subshell  
11189       or separate utility execution environment, such as one of the following:

```

11190 (PATH=foo command -v)
11191 nohup command -v

```

11192 it does not necessarily produce correct results. For example, when called with *nohup* or an *exec*  
 11193 function, in a separate utility execution environment, most implementations are not able to  
 11194 identify aliases, functions, or special built-ins.

11195 Two types of regular built-ins could be encountered on a system and these are described  
 11196 separately by *command*. The description of command search in Section 2.9.1.1 on page 69 allows  
 11197 for a standard utility to be implemented as a regular built-in as long as it is found in the  
 11198 appropriate place in a *PATH* search. So, for example, *command -v true* might yield */bin/true* or  
 11199 some similar path name. Other implementation-dependent utilities that are not defined by this  
 11200 volume of IEEE Std. 1003.1-200x might exist only as built-ins and have no path name associated  
 11201 with them. These produce output identified as (regular) built-ins. Applications encountering  
 11202 these are not able to count on *execing* them, using them with *nohup*, overriding them with a  
 11203 different *PATH*, and so on.

#### 11204 EXAMPLES

- 11205 1. Make a version of *cd* that always prints out the new working directory exactly once:

```

11206 cd() {
11207 command cd "$@" >/dev/null
11208 pwd
11209 }

```

- 11210 2. Start off a “secure shell script” in which the script avoids being spoofed by its parent:

```

11211 IFS='
11212 '
11213 # The preceding value should be <space><tab><newline>.
11214 # Set IFS to its default value.

11215 \unalias -a
11216 # Unset all possible aliases.
11217 # Note that unalias is escaped to prevent an alias
11218 # being used for unalias.

11219 unset -f command
11220 # Ensure command is not a user function.

11221 PATH="$ (command -p getconf _CS_PATH) : $PATH"
11222 # Put on a reliable PATH prefix.

11223 ...

```

11224 At this point, given correct permissions on the directories called by *PATH*, the script has  
 11225 the ability to ensure that any utility it calls is the intended one. It is being very cautious  
 11226 because it assumes that implementation extensions may be present that would allow user  
 11227 functions to exist when it is invoked; this capability is not specified by this volume of  
 11228 IEEE Std. 1003.1-200x, but it is not prohibited as an extension. For example, the *ENV*  
 11229 variable precedes the invocation of the script with a user start-up script. Such a script  
 11230 could define functions to spoof the application.

#### 11231 RATIONALE

11232 Since *command* is a regular built-in utility it is always found prior to the *PATH* search.

11233 There is nothing in the description of *command* that implies the command line is parsed any  
 11234 differently from that of any other simple command. For example:

11235 `command a | b ; c`

11236 is not parsed in any special way that causes ' | ' or ' ; ' to be treated other than a pipe operator  
11237 or semicolon or that prevents function lookup on **b** or **c**.

11238 The *command* utility is somewhat similar to the Eighth Edition shell *builtin* command, but since  
11239 *command* also goes to the file system to search for utilities, the name *builtin* would not be  
11240 intuitive.

11241 The *command* utility is most likely to be provided as a regular built-in. It is not listed as a special  
11242 built-in for the following reasons:

- 11243 • The removal of exportable functions made the special precedence of a special built-in  
11244 unnecessary.
- 11245 • A special built-in has special properties (see Section 2.14 on page 96) that were inappropriate  
11246 for invoking other utilities. For example, two commands such as:

11247 `date > unwritable-file`

11248 `command date > unwritable-file`

11249 would have entirely different results; in a non-interactive script, the former would continue  
11250 to execute the next command, the latter would abort. Introducing this semantic difference  
11251 along with suppressing functions was seen to be non-intuitive.

11252 The **-p** option is present because it is useful to be able to ensure a safe path search that finds all  
11253 the POSIX Shell and Utilities standard utilities. This search might not be identical to the one that  
11254 occurs through one of the POSIX System Interfaces *exec* functions when *PATH* is unset. At the  
11255 very least, this feature is required to allow the script to access the correct version of *getconf* so  
11256 that the value of the default path can be accurately retrieved.

11257 The *command* **-v** and **-V** options were added to satisfy requirements from users that are  
11258 currently accomplished by three different historical utilities: *type* in the System V shell, *whence* in  
11259 the KornShell, and *which* in the C shell. Since there is no historical agreement on how and what  
11260 to accomplish here, the POSIX *command* utility was enhanced and the historical utilities were left  
11261 unmodified. The C shell *which* merely conducts a path search. The KornShell *whence* is more  
11262 elaborate—in addition to the categories required by POSIX, it also reports on tracked aliases,  
11263 exported aliases, and undefined functions.

11264 The output format of **-V** was left mostly unspecified because human users are its only audience.  
11265 Applications should not be written to care about this information; they can use the output of **-v**  
11266 to differentiate between various types of commands, but the additional information that may be  
11267 emitted by the more verbose **-V** is not needed and should not be arbitrarily constrained in its  
11268 verbosity or localization for application parsing reasons.

#### 11269 **FUTURE DIRECTIONS**

11270 None.

#### 11271 **SEE ALSO**

11272 *sh*, *type*

#### 11273 **CHANGE HISTORY**

11274 First released in Issue 4.

11275 **NAME**11276 `compress` — compress data11277 **SYNOPSIS**11278 xSI `compress [-fv][-b bits][file ...]`11279 `compress [-cfv][-b bits][file]`

11280

11281 **DESCRIPTION**

11282 The *compress* utility shall attempt to reduce the size of the named files by using adaptive  
 11283 Lempel-Ziv coding. Except when the output is to the standard output, each file shall be replaced  
 11284 by one with the extension *.Z*. If the invoking process has appropriate privileges, the ownership,  
 11285 modes, access time, and modification time of the original file are preserved. If appending the *.Z*  
 11286 to the file name would make the name exceed {NAME\_MAX} bytes, the command shall fail. If  
 11287 no files are specified, the standard input shall be compressed to the standard output.

11288 **OPTIONS**

11289 The *compress* utility shall conform to the System Interface Definitions volume of  
 11290 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

11291 The following options shall be supported:

11292 **-b *bits*** Specify the maximum number of bits to use in a code. For a portable application,  
 11293 the *bits* argument shall be:

11294  $9 \leq bits \leq 14$

11295 The implementation may allow *bits* values of greater than 14. The default is 14, 15,  
 11296 or 16.

11297 **-c** Cause *compress* to write to the standard output; the input file is not changed, and  
 11298 no *.Z* files are created.

11299 **-f** Force compression of *file*, even if it does not actually reduce the size of the file, or if  
 11300 the corresponding *file.Z* file already exists. If the **-f** option is not given, and the  
 11301 process is not running in the background, the user is prompted as to whether an  
 11302 existing *file.Z* file should be overwritten.

11303 **-v** Write the percentage reduction of each file to standard error.

11304 **OPERANDS**

11305 The following operand shall be supported:

11306 *file* A path name of a file to be compressed.

11307 **STDIN**

11308 The standard input shall be used only if no *file* operands are specified, or if a *file* operand is '- '.

11309 **INPUT FILES**

11310 If *file* operands are specified, the input files contain the data to be compressed.

11311 **ENVIRONMENT VARIABLES**

11312 The following environment variables shall affect the execution of *compress*:

11313 **LANG** Provide a default value for the internationalization variables that are unset or null.  
 11314 If *LANG* is unset or null, the corresponding value from the implementation-  
 11315 dependent default locale shall be used. If any of the internationalization variables  
 11316 contains an invalid setting, the utility shall behave as if none of the variables had  
 11317 been defined.

11318 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 11319 internationalization variables.

11320 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 11321 characters (for example, single-byte as opposed to multi-byte characters in  
 11322 arguments).

11323 *LC\_MESSAGES*  
 11324 Determine the locale that should be used to affect the format and contents of  
 11325 diagnostic messages written to standard error.

11326 *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

11327 **ASYNCHRONOUS EVENTS**  
 11328 Default.

11329 **STDOUT**  
 11330 If no *file* operands are specified, or if a *file* operand is '-', or if the *-c* option is specified, the  
 11331 standard output contains the compressed output.

11332 **STDERR**  
 11333 Used for all diagnostic and prompt messages and the output from *-v*.

11334 **OUTPUT FILES**  
 11335 The output files shall contain the compressed output.

11336 **EXTENDED DESCRIPTION**  
 11337 None.

11338 **EXIT STATUS**  
 11339 The following exit values shall be returned:

11340 0 Successful completion.

11341 1 An error occurred.

11342 2 One or more files were not compressed because they would have increased in size (and the  
 11343 *-f* option was not specified).

11344 >2 An error occurred.

11345 **CONSEQUENCES OF ERRORS**  
 11346 The input file shall remain unmodified.

11347 **APPLICATION USAGE**  
 11348 The amount of compression obtained depends on the size of the input, the number of *bits* per  
 11349 code, and the distribution of common substrings. Typically, text such as source code or English  
 11350 is reduced by 50-60%. Compression is generally much better than that achieved by Huffman  
 11351 coding or adaptive Huffman coding (*compact*), and takes less time to compute.

11352 Although *compress* strictly follows the default actions upon receipt of a signal or when an error  
 11353 occurs, some unexpected results may occur. In some implementations it is likely that a partially  
 11354 compressed file is left in place, alongside its uncompressed input file. Since the general  
 11355 operation of *compress* is to delete the uncompressed file only after the *.Z* file has been  
 11356 successfully filled, an application should always carefully check the exit status of *compress* before  
 11357 arbitrarily deleting files that have like-named neighbors with *.Z* suffixes.

11358 Compressed files are not necessarily portable to other systems.

11359 The limit of 14 on the *bits* option-argument is to achieve portability to all systems (within the  
 11360 restrictions imposed by the lack of an explicit published file format). Some systems based on



11361 16-bit architectures cannot support 15 or 16-bit uncompression.

11362 **EXAMPLES**

11363 None.

11364 **RATIONALE**

11365 None.

11366 **FUTURE DIRECTIONS**

11367 None.

11368 **SEE ALSO**

11369 *uncompress, zcat*

11370 **CHANGE HISTORY**

11371 First released in Issue 4.

11372 **Issue 4, Version 2**

11373 The DESCRIPTION section is clarified to state that the ownership, modes, access time, and  
11374 modification time of the original file are preserved if the invoking process has appropriate  
11375 privileges.

11376 The STDOUT section includes the case where a *file* operand is ' - '.

11377 **Issue 6**

11378 The normative text is reworded to avoid use of the term “must” for application requirements.

## 11379 NAME

11380 cp — copy files

## 11381 SYNOPSIS

11382 cp [-fip] *source\_file target\_file*11383 cp [-fip] *source\_file ... target*11384 cp -R [-H | -L | -P][-fip] *source\_file ... target*11385 cp -r [-H | -L | -P][-fip] *source\_file ... target*

## 11386 DESCRIPTION

11387 The first synopsis form is denoted by two operands, neither of which are existing files of type  
 11388 directory. The *cp* utility shall copy the contents of *source\_file* (or, if *source\_file* is a file of type  
 11389 symbolic link, the contents of the file referenced by *source\_file*) to the destination path named by  
 11390 *target\_file*.

11391 The second synopsis form is denoted by two or more operands where the **-R** or **-r** options are  
 11392 not specified and the first synopsis form is not applicable. It shall be an error if any *source\_file* is a  
 11393 file of type directory, if *target* does not exist, or if *target* is a file of a type defined by the System  
 11394 Interfaces volume of IEEE Std. 1003.1-200x, but is not a file of type directory. The *cp* utility shall  
 11395 copy the contents of each *source\_file* (or, if *source\_file* is a file of type symbolic link, the contents  
 11396 of the file referenced by *source\_file*) to the destination path named by the concatenation of *target*,  
 11397 a slash character, and the last component of *source\_file*.

11398 The third and fourth synopsis forms are denoted by two or more operands where the **-R** or **-r**  
 11399 options are specified. The *cp* utility shall copy each file in the file hierarchy rooted in each  
 11400 *source\_file* to a destination path named as follows.

11401 If *target* exists and is a file of type directory, the name of the corresponding destination path for  
 11402 each file in the file hierarchy shall be the concatenation of *target*, a slash character, and the path  
 11403 name of the file relative to the directory containing *source\_file*.

11404 If *target* does not exist and two operands are specified, the name of the corresponding  
 11405 destination path for *source\_file* shall be *target*; the name of the corresponding destination path for  
 11406 all other files in the file hierarchy shall be the concatenation of *target*, a slash character, and the  
 11407 path name of the file relative to *source\_file*.

11408 It shall be an error if *target* does not exist and more than two operands are specified, or if *target*  
 11409 exists and is a file of a type defined by the System Interfaces volume of IEEE Std. 1003.1-200x,  
 11410 but is not a file of type directory.

11411 In the following description, the term *dest\_file* refers to the file named by the destination path.  
 11412 The term *source\_file* refers to the file that is being copied, whether specified as an operand or a  
 11413 file in a file hierarchy rooted in a *source\_file* operand. If *source\_file* is a file of type symbolic link:

- 11414 • If neither the **-R** nor **-r** options were specified, *cp* shall take actions based on the type and  
 11415 contents of the file referenced by the symbolic link, and not by the symbolic link itself.
- 11416 • If the **-R** option was specified:
  - 11417 — If none of the options **-H**, **-L**, nor **-P** were specified, it is unspecified which of **-H**, **-L**, or  
 11418 **-P** will be used as a default.
  - 11419 — If the **-H** option was specified, *cp* shall take actions based on the type and contents of the  
 11420 file referenced by any symbolic link specified as a *source\_file* operand.
  - 11421 — If the **-L** option was specified, *cp* shall take actions based on the type and contents of the  
 11422 file referenced by any symbolic link specified as a *source\_file* operand or any symbolic

11423 links encountered during traversal of a file hierarchy.

11424 — If the **-P** option was specified, *cp* shall copy any symbolic link specified as a *source\_file*  
 11425 operand and any symbolic links encountered during traversal of a file hierarchy, and shall  
 11426 not follow any symbolic links.

11427 • If the **-r** option was specified, the behavior is implementation-dependent.

11428 For each *source\_file*, the following steps shall be taken:

11429 1. If *source\_file* references the same file as *dest\_file*, *cp* may write a diagnostic message to  
 11430 standard error; it shall do nothing more with *source\_file* and shall go on to any remaining  
 11431 files.

11432 2. If *source\_file* is of type directory, the following steps shall be taken:

11433 a. If neither the **-R** or **-r** options were specified, *cp* shall write a diagnostic message to  
 11434 standard error, do nothing more with *source\_file*, and go on to any remaining files.

11435 b. If *source\_file* was not specified as an operand and *source\_file* is dot or dot-dot, *cp* shall  
 11436 do nothing more with *source\_file* and go on to any remaining files.

11437 c. If *dest\_file* exists and it is a file type not specified by the System Interfaces volume of  
 11438 IEEE Std. 1003.1-200x, the behavior is implementation-dependent.

11439 d. If *dest\_file* exists and it is not of type directory, *cp* shall write a diagnostic message to  
 11440 standard error, do nothing more with *source\_file* or any files below *source\_file* in the  
 11441 file hierarchy, and go on to any remaining files.

11442 e. If the directory *dest\_file* does not exist, it shall be created with file permission bits set  
 11443 to the same value as those of *source\_file*, modified by the file creation mask of the  
 11444 user if the **-p** option was not specified, and then bitwise-inclusively OR'ed with  
 11445 S\_IRWXU. If *dest\_file* cannot be created, *cp* shall write a diagnostic message to  
 11446 standard error, do nothing more with *source\_file*, and go on to any remaining files. It  
 11447 is unspecified if *cp* attempts to copy files in the file hierarchy rooted in *source\_file*.

11448 f. The files in the directory *source\_file* shall be copied to the directory *dest\_file*, taking  
 11449 the four steps [1-4] listed here with the files as *source\_files*.

11450 g. If *dest\_file* was created, its file permission bits shall be changed (if necessary) to be the  
 11451 same as those of *source\_file*, modified by the file creation mask of the user if the **-p**  
 11452 option was not specified.

11453 h. The *cp* utility shall do nothing more with *source\_file* and go on to any remaining files.

11454 3. If *source\_file* is of type regular file, the following steps shall be taken:

11455 a. If *dest\_file* exists, the following steps shall be taken:

11456 i. If the **-i** option is in effect, the *cp* utility shall write a prompt to the standard  
 11457 error and read a line from the standard input. If the response is not affirmative,  
 11458 *cp* shall do nothing more with *source\_file* and go on to any remaining files.

11459 ii. A file descriptor for *dest\_file* shall be obtained by performing actions equivalent  
 11460 to the *open()* function defined in the System Interfaces volume of  
 11461 IEEE Std. 1003.1-200x called using *dest\_file* as the *path* argument, and the  
 11462 bitwise-inclusive OR of O\_WRONLY and O\_TRUNC as the *oflag* argument.

11463 iii. If the attempt to obtain a file descriptor fails and the **-f** option is in effect, *cp*  
 11464 shall attempt to remove the file by performing actions equivalent to the  
 11465 *unlink()* function defined in the System Interfaces volume of

11466 IEEE Std. 1003.1-200x called using *dest\_file* as the *path* argument. If this attempt  
 11467 succeeds, *cp* shall continue with step 3b.

11468 b. If *dest\_file* does not exist, a file descriptor shall be obtained by performing actions  
 11469 equivalent to the *open()* function defined in the System Interfaces volume of  
 11470 IEEE Std. 1003.1-200x called using *dest\_file* as the *path* argument, and the bitwise-  
 11471 inclusive OR of O\_WRONLY and O\_CREAT as the *oflag* argument. The file  
 11472 permission bits of *source\_file* shall be the *mode* argument.

11473 c. If the attempt to obtain a file descriptor fails, *cp* shall write a diagnostic message to  
 11474 standard error, do nothing more with *source\_file*, and go on to any remaining files.

11475 d. The contents of *source\_file* shall be written to the file descriptor. Any write errors  
 11476 shall cause *cp* to write a diagnostic message to standard error and continue to step 3e.

11477 e. The file descriptor shall be closed.

11478 f. The *cp* utility shall do nothing more with *source\_file*. If a write error occurred in step  
 11479 3d, it is unspecified if *cp* continues with any remaining files. If no write error  
 11480 occurred in step 3d, *cp* shall go on to any remaining files.

11481 4. Otherwise, the following steps shall be taken:

11482 a. If the **-r** option was specified, the behavior is implementation-dependent.

11483 b. If the **-R** option was specified, the following steps shall be taken:

11484 i. The *dest\_file* shall be created with the same file type as *source\_file*.

11485 ii. If *source\_file* is a file of type FIFO, the file permission bits shall be the same as  
 11486 those of *source\_file*, modified by the file creation mask of the user if the **-p**  
 11487 option was not specified. Otherwise, the permissions, owner ID, and group ID  
 11488 of *dest\_file* are implementation-dependent.

11489 If this creation fails for any reason, *cp* shall write a diagnostic message to  
 11490 standard error, do nothing more with *source\_file*, and go on to any remaining  
 11491 files.

11492 iii. If *source\_file* is a file of type symbolic link, the path name contained in *dest\_file*  
 11493 shall be the same as the path name contained in *source\_file*.

11494 If this fails for any reason, *cp* shall write a diagnostic message to standard error,  
 11495 do nothing more with *source\_file*, and go on to any remaining files.

11496 If the implementation provides additional or alternate access control mechanisms (see the  
 11497 System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 4.1, File Access  
 11498 Permissions), their effect on copies of files is implementation-dependent.

#### 11499 OPTIONS

11500 The *cp* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 11501 Section 12.2, Utility Syntax Guidelines.

11502 The following options shall be supported:

11503 **-f** If a file descriptor for a destination file cannot be obtained, as described in step  
 11504 3.a.ii., attempt to unlink the destination file and proceed.

11505 **-H** Take actions based on the type and contents of the file referenced by any symbolic  
 11506 link specified as a *source\_file* operand.

11507 **-i** Write a prompt to standard error before copying to any existing destination file. If  
 11508 the response from the standard input is affirmative, the copy shall be attempted;

11509 otherwise, it shall not.

11510       **-L**       Take actions based on the type and contents of the file referenced by any symbolic  
11511 link specified as a *source\_file* operand or any symbolic links encountered during  
11512 traversal of a file hierarchy.

### 11513 **Notes to Reviewers**

11514 *This section with side shading will not appear in the final copy. - Ed.*

11515 A description of the **-P** option is needed. This is not in the IEEE P1003.2b draft  
11516 standard.

11517       **-p**       Duplicate the following characteristics of each source file in the corresponding  
11518 destination file:

- 11519           1. The time of last data modification and time of last access. If this duplication  
11520           fails for any reason, *cp* shall write a diagnostic message to standard error.
- 11521           2. The user ID and group ID. If this duplication fails for any reason, it is  
11522           unspecified whether *cp* writes a diagnostic message to standard error.
- 11523           3. The file permission bits and the S\_ISUID and S\_ISGID bits. Other,  
11524           implementation-dependent, bits may be duplicated as well. If this  
11525           duplication fails for any reason, *cp* shall write a diagnostic message to  
11526           standard error.

11527 If the user ID or the group ID cannot be duplicated, the file permission bits  
11528 S\_ISUID and S\_ISGID shall be cleared. If these bits are present in the source file but  
11529 are not duplicated in the destination file, it is unspecified whether *cp* writes a  
11530 diagnostic message to standard error.

11531 The order in which the preceding characteristics are duplicated is unspecified. The  
11532 *dest\_file* shall not be deleted if these characteristics cannot be preserved.

11533       **-R**       Copy file hierarchies.

11534 OB       **-r**       Copy file hierarchies. The treatment of special files is implementation-dependent.

11535 Specifying more than one of the mutually-exclusive options **-H**, **-L**, and **-P** shall not be  
11536 considered an error. The last option specified shall determine the behavior of the utility.

### 11537 **OPERANDS**

11538 The following operands shall be supported:

11539       *source\_file*   A path name of a file to be copied.

11540       *target\_file*   A path name of an existing or nonexistent file, used for the output when a single  
11541 file is copied.

11542       *target*       A path name of a directory to contain the copied files.

### 11543 **STDIN**

11544 Used to read an input line in response to each prompt specified in the STDERR section.  
11545 Otherwise, the standard input shall not be used.

### 11546 **INPUT FILES**

11547 The input files specified as operands may be of any file type.

11548 **ENVIRONMENT VARIABLES**

11549 The following environment variables shall affect the execution of *cp*:

11550 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 11551 If *LANG* is unset or null, the corresponding value from the implementation-  
 11552 dependent default locale shall be used. If any of the internationalization variables  
 11553 contains an invalid setting, the utility shall behave as if none of the variables had  
 11554 been defined.

11555 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 11556 internationalization variables.

11557 *LC\_COLLATE*  
 11558 Determine the locale for the behavior of ranges, equivalence classes, and multi-  
 11559 character collating elements used in the extended regular expression defined for  
 11560 the **yesexpr** locale keyword in the *LC\_MESSAGES* category.

11561 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 11562 characters (for example, single-byte as opposed to multi-byte characters in  
 11563 arguments and input files) and the behavior of character classes used in the  
 11564 extended regular expression defined for the **yesexpr** locale keyword in the  
 11565 *LC\_MESSAGES* category.

11566 *LC\_MESSAGES*  
 11567 Determine the locale for the processing of affirmative responses that should be  
 11568 used to affect the format and contents of diagnostic messages written to standard  
 11569 error.

11570 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

11571 **ASYNCHRONOUS EVENTS**

11572 Default.

11573 **STDOUT**

11574 Not used.

11575 **STDERR**

11576 A prompt shall be written to standard error under the conditions specified in the DESCRIPTION  
 11577 section. The prompt shall contain the destination path name, but its format is otherwise  
 11578 unspecified. Otherwise, the standard error shall be used only for diagnostic messages.

11579 **OUTPUT FILES**

11580 The output files may be of any type.

11581 **EXTENDED DESCRIPTION**

11582 None.

11583 **EXIT STATUS**

11584 The following exit values shall be returned:

11585 0 All files were copied successfully.

11586 >0 An error occurred.

11587 **CONSEQUENCES OF ERRORS**

11588 If *cp* is prematurely terminated by a signal or error, files or file hierarchies may be only partially  
 11589 copied and files and directories may have incorrect permissions or access and modification  
 11590 times.

11591 **APPLICATION USAGE**

11592 The difference between **-R** and **-r** is in the treatment by *cp* of file types other than regular and  
11593 directory. The original **-r** flag, for historic reasons, does not handle special files any differently  
11594 from regular files, but always reads the file and copies its contents. This has obvious problems in  
11595 the presence of special file types; for example, character devices, FIFOs, and sockets. The **-R**  
11596 option is intended to recreate the file hierarchy and the **-r** option supports historical practice. It  
11597 was anticipated that a future version of this volume of IEEE Std. 1003.1-200x would deprecate  
11598 the **-r** option, and for that reason, there has been no attempt to fix its behavior with respect to  
11599 FIFOs or other file types where copying the file is clearly wrong. However, some systems  
11600 support **-r** with the same abilities as the **-R** defined in this volume of IEEE Std. 1003.1-200x. To  
11601 accommodate them as well as systems that do not, the differences between **-r** and **-R** are  
11602 implementation-dependent. Implementations may make them identical. The **-r** option is now  
11603 marked obsolescent.

11604 The set-user-ID and set-group-ID bits are explicitly cleared when files are created. This is to  
11605 prevent users from creating programs that are set-user-ID or set-group-ID to them when  
11606 copying files or to make set-user-ID or set-group-ID files accessible to new groups of users. For  
11607 example, if a file is set-user-ID and the copy has a different group ID than the source, a new  
11608 group of users has execute permission to a set-user-ID program than did previously. In  
11609 particular, this is a problem for superusers copying users' trees.

11610 **EXAMPLES**

11611 None.

11612 **RATIONALE**

11613 The **-i** option exists on BSD systems, giving applications and users a way to avoid accidentally  
11614 removing files when copying. Although the 4.3 BSD version does not prompt if the standard  
11615 input is not a terminal, the standard developers decided that use of **-i** is a request for interaction,  
11616 so when the destination path exists, the utility takes instructions from whatever responds on  
11617 standard input.

11618 The exact format of the interactive prompts is unspecified. Only the general nature of the  
11619 contents of prompts are specified because implementations may desire more descriptive  
11620 prompts than those used on historical implementations. Therefore, an application using the **-i**  
11621 option relies on the system to provide the most suitable dialog directly with the user, based on  
11622 the behavior specified.

11623 The **-p** option is historical practice on BSD systems, duplicating the time of last data  
11624 modification and time of last access. This volume of IEEE Std. 1003.1-200x extends it to preserve  
11625 the user and group IDs, as well as the file permissions. This requirement has obvious problems  
11626 in that the directories are almost certainly modified after being copied. This volume of  
11627 IEEE Std. 1003.1-200x requires that the modification times be preserved. The statement that the  
11628 order in which the characteristics are duplicated is unspecified is to permit implementations to  
11629 provide the maximum amount of security for the user. Implementations should take into  
11630 account the obvious security issues involved in setting the owner, group, and mode in the  
11631 wrong order or creating files with an owner, group, or mode different from the final value.

11632 It is unspecified whether *cp* writes diagnostic messages when the user and group IDs cannot be  
11633 set due to the widespread practice of users using **-p** to duplicate some portion of the file  
11634 characteristics, indifferent to the duplication of others. Historic implementations only write  
11635 diagnostic messages on errors other than [EPERM].

11636 The **-r** option is historical practice on BSD and BSD-derived systems, copying file hierarchies as  
11637 opposed to single files. This functionality is used heavily in historical applications, and its loss  
11638 would significantly decrease consensus. The **-R** option was added as a close synonym to the **-r**  
11639 option, selected for consistency with all other options in this volume of IEEE Std. 1003.1-200x

11640 that do recursive directory descent.

11641 When a failure occurs during the copying of a file hierarchy, *cp* is required to attempt to copy  
11642 files that are on the same level in the hierarchy or above the file where the failure occurred. It is  
11643 unspecified if *cp* shall attempt to copy files below the file where the failure occurred (which  
11644 cannot succeed in any case).

11645 Permissions, owners, and groups of created special file types have been deliberately left as  
11646 implementation-dependent. This is to allow systems to satisfy special requirements (for  
11647 example, allowing users to create character special devices, but requiring them to be owned by a  
11648 certain group). In general, it is strongly suggested that the permissions, owner, and group be the  
11649 same as if the user had run the historical *mknod*, *ln*, or other utility to create the file. It is also  
11650 probable that additional privileges are required to create block, character, or other  
11651 implementation-dependent special file types.

11652 Additionally, the *-p* option explicitly requires that all set-user-ID and set-group-ID permissions  
11653 be discarded if any of the owner or group IDs cannot be set. This is to keep users from  
11654 unintentionally giving away special privilege when copying programs.

11655 When creating regular files, historical versions of *cp* use the mode of the source file as modified  
11656 by the file mode creation mask. Other choices would have been to use the mode of the source file  
11657 unmodified by the creation mask or to use the same mode as would be given to a new file  
11658 created by the user (plus the execution bits of the source file) and then modify it by the file mode  
11659 creation mask. In the absence of any strong reason to change historic practice, it was in large part  
11660 retained.

11661 When creating directories, historical versions of *cp* use the mode of the source directory, plus  
11662 read, write, and search bits for the owner, as modified by the file mode creation mask. This is  
11663 done so that *cp* can copy trees where the user has read permission, but the owner does not. A  
11664 side effect is that if the file creation mask denies the owner permissions, *cp* fails. Also, once the  
11665 copy is done, historical versions of *cp* set the permissions on the created directory to be the same  
11666 as the source directory, unmodified by the file creation mask.

11667 This behavior has been modified so that *cp* is always able to create the contents of the directory,  
11668 regardless of the file creation mask. After the copy is done, the permissions are set to be the same  
11669 as the source directory, as modified by the file creation mask. This latter change from historical  
11670 behavior is to prevent users from accidentally creating directories with permissions beyond  
11671 those they would normally set and for consistency with the behavior of *cp* in creating files.

11672 It is not a requirement that *cp* detect attempts to copy a file to itself; however, implementations  
11673 are strongly encouraged to do so. Historical implementations have detected the attempt in most  
11674 cases.

11675 There are two methods of copying subtrees in this volume of IEEE Std. 1003.1-200x. The other  
11676 method is described as part of the *pax* utility (see *pax* on page 737). Both methods are historical  
11677 practice. The *cp* utility provides a simpler, more intuitive interface, while *pax* offers a finer  
11678 granularity of control. Each provides additional functionality to the other; in particular, *pax*  
11679 maintains the hard-link structure of the hierarchy, while *cp* does not. It is the intention of the  
11680 standard developers that the results be similar (using appropriate option combinations in both  
11681 utilities). The results are not required to be identical; there seemed insufficient gain to  
11682 applications to balance the difficulty of implementations having to guarantee that the results  
11683 would be exactly identical.

11684 The wording allowing *cp* to copy a directory to implementation-dependent file types not  
11685 specified by the System Interfaces volume of IEEE Std. 1003.1-200x is provided so that  
11686 implementations supporting symbolic links are not required to prohibit copying directories to  
11687 symbolic links. Other extensions to the System Interfaces volume of IEEE Std. 1003.1-200x file



11688 types may need to use this loophole as well.

11689 **FUTURE DIRECTIONS**

11690 The **-r** option may be removed; use **-R** instead.

11691 **SEE ALSO**

11692 *mv, find, ln, pax*

11693 **CHANGE HISTORY**

11694 First released in Issue 2.

11695 **Issue 4**

11696 Aligned with the ISO/IEC 9945-2:1993 standard.

11697 **Issue 6**

11698 The **-r** option is marked obsolescent.

11699 The new options **-H**, **-L**, and **-P** are added to align with the IEEE P1003.2b draft standard. These  
11700 options affect the processing of symbolic links.

## 11701 NAME

11702 crontab — schedule periodic background work

## 11703 SYNOPSIS

11704 UP crontab [*file*]

11705 crontab [ -e | -l | -r ]

11706

## 11707 DESCRIPTION

11708 The *crontab* utility shall create, replace, or edit a user's *crontab* entry; a crontab entry is a list of  
 11709 commands and the times at which they shall be executed. The new crontab entry can be input by  
 11710 specifying *file* or input from standard input if no *file* operand is specified, or by using an editor, if  
 11711 **-e** is specified.

11712 Upon execution of a command from a crontab entry, the implementation shall supply a default  
 11713 environment, defining at least the following environment variables:

11714 **HOME** A path name of the user's home directory.

11715 **LOGNAME** The user's login name.

11716 **PATH** A string representing a search path guaranteed to find all of the standard utilities.

11717 **SHELL** A path name of the command interpreter. When *crontab* is invoked as specified by  
 11718 this volume of IEEE Std. 1003.1-200x, the value shall be a path name for *sh*.

11719 The values of these variables when *crontab* is invoked as specified by this volume of  
 11720 IEEE Std. 1003.1-200x shall not affect the default values provided when the scheduled command  
 11721 is run.

11722 If standard output and standard error are not redirected by commands executed from the  
 11723 crontab entry, any generated output or errors shall be mailed, via an implementation-dependent  
 11724 method, to the user.

11725 XSI Users are permitted to use *crontab* if their names appear in the file **/usr/lib/cron/cron.allow**. If  
 11726 that file does not exist, the file **/usr/lib/cron/cron.deny** is checked to determine whether the user  
 11727 should be denied access to *crontab*. If neither file exists, only a process with appropriate  
 11728 privileges is allowed to submit a job. If only **cron.deny** exists and is empty, global usage is  
 11729 permitted. The **cron.allow** and **cron.deny** files consist of one user name per line.

## 11730 OPTIONS

11731 The *crontab* utility shall conform to the System Interface Definitions volume of  
 11732 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

11733 The following options shall be supported:

11734 **-e** Edit a copy of the invoking user's crontab entry, or create an empty entry to edit if  
 11735 the crontab entry does not exist. When editing is complete, the entry shall be  
 11736 installed as the user's crontab entry.

11737 **-l** (The letter ell.) List the invoking user's crontab entry.

11738 **-r** Remove the invoking user's crontab entry.

## 11739 OPERANDS

11740 The following operand shall be supported:

11741 **file** The path name of a file that contains specifications, in the format defined in the  
 11742 INPUT FILES section, for crontab entries.

11743 **STDIN**

11744 See the INPUT FILES section.

11745 **INPUT FILES**

11746 In the POSIX locale, the user or application shall ensure that a crontab entry is a text file  
 11747 consisting of lines of six fields each. The fields shall be separated by <blank> characters. The first  
 11748 five fields shall be integer patterns that specify the following:

- 11749 1. Minute (0-59)
- 11750 2. Hour (0-23)
- 11751 3. Day of the month (1-31)
- 11752 4. Month of the year (1-12)
- 11753 5. Day of the week (0-6 with 0=Sunday)

11754 Each of these patterns can be either an asterisk (meaning all valid values), an element, or a list of  
 11755 elements separated by commas. An element shall be either a number or two numbers separated  
 11756 by a hyphen (meaning an inclusive range). The specification of days can be made by two fields  
 11757 (day of the month and day of the week). If month, day of month, and day of week are all  
 11758 asterisks, every day shall be matched. If either the month or day of month is specified as an  
 11759 element or list, but the day of week is an asterisk, the month and day of month fields shall  
 11760 specify the days that match. If both month and day of month are specified as asterisk, but day of  
 11761 week is an element or list, then only the specified days of the week match. Finally, if either the  
 11762 month or day of month is specified as an element or list, and the day of week is also specified as  
 11763 an element or list, then any day matching either the month and day of month, or the day of  
 11764 week, shall be matched.

11765 The sixth field of a line in a crontab entry is a string that shall be executed by *sh* at the specified  
 11766 times. A percent sign character in this field shall be translated to a <newline> character. Any  
 11767 character preceded by a backslash (including the '%') shall cause that character to be treated  
 11768 literally. Only the first line (up to a '%' or end-of-line) of the command field shall be executed  
 11769 by the command interpreter. The other lines shall be made available to the command as  
 11770 standard input.

11771 Blank lines and those whose first non-<blank> character is '#' shall be ignored.

11772 XSI The text files `/usr/lib/cron/cron.allow` and `/usr/lib/cron/cron.deny` contain user names, one per  
 11773 line, of users who are, respectively, authorized or denied access to the service underlying the  
 11774 *crontab* utility.

11775 **ENVIRONMENT VARIABLES**

11776 The following environment variables shall affect the execution of *crontab*:

- |                                           |                 |                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------------------------------|-----------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 11777<br>11778                            | <b>EDITOR</b>   | Determine the editor to be invoked when the <code>-e</code> option is specified. The default editor shall be <i>vi</i> .                                                                                                                                                                                                                                         |
| 11779<br>11780<br>11781<br>11782<br>11783 | <b>LANG</b>     | Provide a default value for the internationalization variables that are unset or null. If <i>LANG</i> is unset or null, the corresponding value from the implementation-dependent default locale shall be used. If any of the internationalization variables contains an invalid setting, the utility shall behave as if none of the variables had been defined. |
| 11784<br>11785                            | <b>LC_ALL</b>   | If set to a non-empty string value, override the values of all the other internationalization variables.                                                                                                                                                                                                                                                         |
| 11786<br>11787                            | <b>LC_CTYPE</b> | Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in                                                                                                                                                                                                    |

11788 arguments and input files).

11789 **LC\_MESSAGES**

11790 Determine the locale that should be used to affect the format and contents of

11791 diagnostic messages written to standard error.

11792 XSI **NLSPATH** Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

11793 **ASYNCHRONOUS EVENTS**

11794 Default.

11795 **STDOUT**

11796 If the **-l** option is specified, the crontab entry shall be written to the standard output.

11797 **STDERR**

11798 Used only for diagnostic messages.

11799 **OUTPUT FILES**

11800 None.

11801 **EXTENDED DESCRIPTION**

11802 None.

11803 **EXIT STATUS**

11804 The following exit values shall be returned:

11805 0 Successful completion.

11806 >0 An error occurred.

11807 **CONSEQUENCES OF ERRORS**

11808 The user's crontab entry is not submitted, removed, edited, or listed.

11809 **APPLICATION USAGE**

11810 The format of the *crontab* entry shown here is guaranteed only for the POSIX locale. Other

11811 cultures may be supported with substantially different interfaces, although implementations are

11812 encouraged to provide comparable levels of functionality.

11813 The default settings of the *HOME*, *LOGNAME*, *PATH*, and *SHELL* variables that are given to the

11814 scheduled job are not affected by the settings of those variables when *crontab* is run; as stated,

11815 they are defaults. The text about “invoked as specified by this volume of IEEE Std. 1003.1-200x”

11816 means that the implementation may provide extensions that allow these variables to be affected

11817 at runtime, but that the user has to take explicit action in order to access the extension, such as

11818 give a new option flag or modify the format of the crontab entry.

11819 A typical user error is to type only *crontab*; this causes the system to wait for the new crontab

11820 entry on standard input. If end-of-file is typed (generally <control>-D), the crontab entry is

11821 replaced by an empty file. In this case, the user should type the interrupt character, which

11822 prevents the crontab entry from being replaced.

11823 Application writers should note that this utility need not be provided on systems that do not

11824 support the User Portability Utilities option.

11825 **EXAMPLES**

11826 1. Clean up **core** files every weekday morning at 3:15 am:

11827 15 3 \* \* 1-5 find \$HOME -name core 2>/dev/null | xargs rm -f

11828 2. Mail a birthday greeting:

11829           0 12 14 2 \* mailx john%Happy Birthday!%Time for lunch.

11830           3. As an example of specifying the two types of days:

11831           0 0 1,15 \* 1

11832           would run a command on the first and fifteenth of each month, as well as on every  
11833           Monday. To specify days by only one field, the other field should be set to `'*'`; for  
11834           example:

11835           0 0 \* \* 1

11836           would run a command only on Mondays.

#### 11837 **RATIONALE**

11838           All references to a *cron* daemon and to *cron files* have been omitted. Although historical  
11839           implementations have used this arrangement, there is no reason to limit future implementations.

11840           This description of *crontab* is designed to support only users with normal privileges. The format  
11841           of the input is based on the System V *crontab*; however, there is no requirement here that the  
11842           actual system database used by the *cron* daemon (or a similar mechanism) use this format  
11843           internally. For example, systems derived from BSD are likely to have an additional field  
11844           appended that indicates the user identity to be used when the job is submitted.

11845           The `-e` option was adopted from the SVID as a user convenience, although it does not exist in all  
11846           historical implementations.

#### 11847 **FUTURE DIRECTIONS**

11848           None.

#### 11849 **SEE ALSO**

11850           *at*

#### 11851 **CHANGE HISTORY**

11852           First released in Issue 2.

#### 11853 **Issue 4**

11854           Aligned with the ISO/IEC 9945-2:1993 standard.

#### 11855 **Issue 6**

11856           This utility is now marked as part of the User Portability Utilities option. |

11857           The normative text is reworded to avoid use of the term “must” for application requirements. |

11858 **NAME**

11859       csplit — split files based on context

11860 **SYNOPSIS**

11861 UP       csplit [-ks][-f *prefix*][-n *number*] *file* *arg1* ...*argn*

11862

11863 **DESCRIPTION**

11864       The *csplit* utility shall read the file named by the *file* operand, write all or part of that file into  
11865       other files as directed by the *arg* operands, and write the sizes of the files.

11866 **OPTIONS**

11867       The *csplit* utility shall conform to the System Interface Definitions volume of  
11868       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

11869       The following options shall be supported:

11870       **-f *prefix***     Name the created files *prefix00*, *prefix01*, ..., *prefixn*. The default is **xx00** ... **xxn**. If  
11871       the *prefix* argument would create a file name exceeding {NAME\_MAX} bytes, an  
11872       error shall result, *csplit* shall exit with a diagnostic message and no files shall be  
11873       created.

11874       **-k**             Leave previously created files intact. By default, *csplit* shall remove created files if  
11875       an error occurs.

11876       **-n *number***     Use *number* decimal digits to form file names for the file pieces. The default shall be  
11877       2.

11878       **-s**             Suppress the output of file size messages.

11879 **OPERANDS**

11880       The following operands shall be supported:

11881       ***file***           The path name of a text file to be split. If *file* is '-', the standard input shall be  
11882       used.

11883       The operands *arg1* ... *argn* can be a combination of the following:

11884       /*regexp*/[*offset*]

11885       A file shall be created using the content of the lines from the current line up to, but  
11886       not including, the line that results from the evaluation of the regular expression  
11887       with *offset*, if any, applied. The regular expression *regexp* shall follow the rules for  
11888       basic regular expressions described in the System Interface Definitions volume of  
11889       IEEE Std. 1003.1-200x, Section 9.3, Basic Regular Expressions. The application shall  
11890       use the sequence "\/" to specify a slash character within the *regexp*. The optional  
11891       offset shall be a positive or negative integer value representing a number of lines.  
11892       A positive integer value can be preceded by '+'. If the selection of lines from an  
11893       *offset* expression of this type would create a file with zero lines, or one with greater  
11894       than the number of lines left in the input file, the results are unspecified. After the  
11895       section is created, the current line shall be set to the line that results from the  
11896       evaluation of the regular expression with any offset applied. If the current line is  
11897       the first line in the file and a regular expression operation has not yet been  
11898       performed, the pattern match of *regexp* shall be applied from the current line to the  
11899       end of the file. Otherwise, the pattern match of *regexp* shall be applied from the line  
11900       following the current line to the end of the file.

11901       %*regexp*%[*offset*]

11902       Equivalent to /*regexp*/[*offset*], except that no file shall be created for the selected  
11903       section of the input file. The application shall use the sequence "\%" to specify a

|       |                              |                                                                                                              |  |
|-------|------------------------------|--------------------------------------------------------------------------------------------------------------|--|
| 11904 |                              | percent-sign character within the <i>rexp</i> .                                                              |  |
| 11905 | <i>line_no</i>               | Create a file from the current line up to (but not including) the line number <i>line_no</i> .               |  |
| 11906 |                              | Lines in the file shall be numbered starting at one. The current line becomes                                |  |
| 11907 |                              | <i>line_no</i> .                                                                                             |  |
| 11908 | <i>{num}</i>                 | Repeat operand. This operand can follow any of the operands described                                        |  |
| 11909 |                              | previously. If it follows a <i>rexp</i> type operand, that operand shall be applied <i>num</i>               |  |
| 11910 |                              | more times. If it follows a <i>line_no</i> operand, the file shall be split every <i>line_no</i> lines,      |  |
| 11911 |                              | <i>num</i> times, from that point.                                                                           |  |
| 11912 |                              | An error shall be reported if an operand does not reference a line between the current position              |  |
| 11913 |                              | and the end of the file.                                                                                     |  |
| 11914 | <b>STDIN</b>                 |                                                                                                              |  |
| 11915 |                              | See the INPUT FILES section.                                                                                 |  |
| 11916 | <b>INPUT FILES</b>           |                                                                                                              |  |
| 11917 |                              | The input file shall be a text file.                                                                         |  |
| 11918 | <b>ENVIRONMENT VARIABLES</b> |                                                                                                              |  |
| 11919 |                              | The following environment variables shall affect the execution of <i>csplit</i> :                            |  |
| 11920 | <i>LANG</i>                  | Provide a default value for the internationalization variables that are unset or null.                       |  |
| 11921 |                              | If <i>LANG</i> is unset or null, the corresponding value from the implementation-                            |  |
| 11922 |                              | dependent default locale shall be used. If any of the internationalization variables                         |  |
| 11923 |                              | contains an invalid setting, the utility shall behave as if none of the variables had                        |  |
| 11924 |                              | been defined.                                                                                                |  |
| 11925 | <i>LC_ALL</i>                | If set to a non-empty string value, override the values of all the other                                     |  |
| 11926 |                              | internationalization variables.                                                                              |  |
| 11927 | <i>LC_COLLATE</i>            |                                                                                                              |  |
| 11928 |                              | Determine the locale for the behavior of ranges, equivalence classes, and multi-                             |  |
| 11929 |                              | character collating elements within regular expressions.                                                     |  |
| 11930 | <i>LC_CTYPE</i>              | Determine the locale for the interpretation of sequences of bytes of text data as                            |  |
| 11931 |                              | characters (for example, single-byte as opposed to multi-byte characters in                                  |  |
| 11932 |                              | arguments and input files) and the behavior of character classes within regular                              |  |
| 11933 |                              | expressions.                                                                                                 |  |
| 11934 | <i>LC_MESSAGES</i>           |                                                                                                              |  |
| 11935 |                              | Determine the locale that should be used to affect the format and contents of                                |  |
| 11936 |                              | diagnostic messages written to standard error.                                                               |  |
| 11937 | XSI <i>NLSPATH</i>           | Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .                        |  |
| 11938 | <b>ASYNCHRONOUS EVENTS</b>   |                                                                                                              |  |
| 11939 |                              | If the <i>-k</i> option is specified, created files shall be retained. Otherwise, the default action occurs. |  |
| 11940 | <b>STDOUT</b>                |                                                                                                              |  |
| 11941 |                              | Unless the <i>-s</i> option is used, the standard output shall consist of one line per file created, with a  |  |
| 11942 |                              | format as follows:                                                                                           |  |
| 11943 |                              | <i>"%d\n", &lt;file size in bytes&gt;</i>                                                                    |  |

11944 **STDERR**

11945       Used only for diagnostic messages.

11946 **OUTPUT FILES**

11947       The output files shall contain portions of the original input file; otherwise, unchanged.

11948 **EXTENDED DESCRIPTION**

11949       None.

11950 **EXIT STATUS**

11951       The following exit values shall be returned:

11952       0   Successful completion.

11953       &gt;0   An error occurred.

11954 **CONSEQUENCES OF ERRORS**11955       By default, created files shall be removed if an error occurs. When the **-k** option is specified,  
11956       created files shall not be removed if an error occurs.11957 **APPLICATION USAGE**11958       Application writers should note that this utility need not be provided on systems that do not  
11959       support the User Portability Utilities option.11960 **EXAMPLES**11961       1. This example creates four files, **cobol00 ... cobol03**:11962       `csplit -f cobol file '/procedure division/' /par5./ /par16./`

11963       After editing the split files, they can be recombined as follows:

11964       `cat cobol0[0-3] > file`

11965       Note that this example overwrites the original file.

11966       2. This example would split the file after the first 99 lines, and every 100 lines thereafter, up  
11967       to 9999 lines; this is because lines in the file are numbered from 1 rather than zero, for  
11968       historical reasons:11969       `csplit -k file 100 {99}`11970       3. Assuming that **prog.c** follows the C-language coding convention of ending routines with a  
11971       ' } ' at the beginning of the line, this example creates a file containing each separate C  
11972       routine (up to 21) in **prog.c**:11973       `csplit -k prog.c '%main(%' '/^}/+1' {20}`11974 **RATIONALE**11975       The **-n** option was added to extend the range of file names that could be handled.11976       Consideration was given to adding a **-a** flag to use the alphabetic file name generation used by  
11977       the historical *split* utility, but the functionality added by the **-n** option was deemed to make  
11978       alphabetic naming unnecessary.11979 **FUTURE DIRECTIONS**

11980       None.

11981 **SEE ALSO**11982       *sed*, *split*



11983 **CHANGE HISTORY**

11984 First released in Issue 2.

11985 **Issue 4**

11986 Aligned with the ISO/IEC 9945-2: 1993 standard.

11987 **Issue 5**

11988 FUTURE DIRECTIONS section added.

11989 **Issue 6**

11990 This utility is now marked as part of the User Portability Utilities option.

11991 The APPLICATION USAGE section is added.

11992 The description of regular expression operands is changed to align with the IEEE P1003.2b draft  
11993 standard.

11994 The normative text is reworded to avoid use of the term “must” for application requirements.

11995 **NAME**11996       ctags — create a tags file (**DEVELOPMENT**, **FORTRAN**)11997 **SYNOPSIS**11998 UP       ctags [-a][-f *tagsfile*] *pathname* ...11999       ctags -x *pathname* ...

12000

12001 **DESCRIPTION**

12002       The *ctags* utility shall be provided on systems that support the User Portability Utilities option,  
 12003       the Software Development Utilities option, and either or both of the C-Language Development  
 12004       Utilities option and FORTRAN Development Utilities option. On other systems, it is optional.

12005       The *ctags* utility shall create a *tags* file or an index of objects from C-language or FORTRAN  
 12006       source files specified by the *pathname* operands. The tags file shall list the locators of language-  
 12007       specific objects within the source files. A locator consists of a name, path name, and either a  
 12008       search pattern or a line number that can be used in searching for the object definition. The  
 12009       objects that shall be recognized are specified in the EXTENDED DESCRIPTION section.

12010 **OPTIONS**

12011       The *ctags* utility shall conform to the System Interface Definitions volume of  
 12012       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

12013       The following options shall be supported:

12014       **-a**           Append to tags file.

12015       **-f *tagsfile***   Write the object locator lists into *tagsfile* instead of the default file named **tags** in  
 12016       the current directory.

12017       **-x**           Produce a list of object names, the line number, and file name in which each is  
 12018       defined, as well as the text of that line, and write this to the standard output. A  
 12019       **tags** file shall not be created when **-x** is specified.

12020 **OPERANDS**12021       The following *pathname* operands are supported:

12022       ***file.c***       Files with basenames ending with the **.c** suffix shall be treated as C-language  
 12023       source code. Such files that are not valid input to *c89* produce unspecified results.

12024       ***file.h***       Files with basenames ending with the **.h** suffix shall be treated as C-language  
 12025       source code. Such files that are not valid input to *c89* produce unspecified results.

12026       ***file.f***       Files with basenames ending with the **.f** suffix shall be treated as FORTRAN-  
 12027       language source code. Such files that are not valid input to *fort77* produce  
 12028       unspecified results.

12029       The handling of other files is implementation-dependent.

12030 **STDIN**

12031       See the INPUT FILES section.

12032 **INPUT FILES**

12033       The input files shall be text files containing source code in the language indicated by the operand  
 12034       file name suffixes.

12035 **ENVIRONMENT VARIABLES**

12036 The following environment variables shall affect the execution of *ctags*:

12037 **LANG** Provide a default value for the internationalization variables that are unset or null.  
 12038 If *LANG* is unset or null, the corresponding value from the implementation-  
 12039 dependent default locale shall be used. If any of the internationalization variables  
 12040 contains an invalid setting, the utility shall behave as if none of the variables had  
 12041 been defined.

12042 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 12043 internationalization variables.

12044 **LC\_COLLATE**  
 12045 Determine the order in which output is sorted for the *-x* option. The POSIX locale  
 12046 determines the order in which the tags file is written.

12047 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 12048 characters (for example, single-byte as opposed to multi-byte characters in  
 12049 arguments and input files). When processing C-language source code, if the locale  
 12050 is not compatible with the C locale described by the ISO C standard, the results are  
 12051 unspecified.

12052 **LC\_MESSAGES**  
 12053 Determine the locale that should be used to affect the format and contents of  
 12054 diagnostic messages written to standard error.

12055 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

12056 **ASYNCHRONOUS EVENTS**

12057 Default.

12058 **STDOUT**

12059 The list of object name information produced by the *-x* option shall be written to standard  
 12060 output in the following format:

12061 "%s %d %s %s", <object-name>, <line-number>, <filename>,  
 12062 <text>

12063 where <text> is the text of line <line-number> of file <filename>.

12064 **STDERR**

12065 Used only for diagnostic messages.

12066 **OUTPUT FILES**

12067 When the *-x* option is not specified, the format of the output file shall be:

12068 "%s\t%s\t/%s/\n", <identifier>, <filename>, <pattern>

12069 where <pattern> is a search pattern that could be used by an editor to find the defining instance  
 12070 of <identifier> in <filename> (where *defining instance* is indicated by the declarations listed in the  
 12071 EXTENDED DESCRIPTION).

12072 An optional circumflex ('^') can be added as a prefix to <pattern>, and an optional dollar sign  
 12073 can be appended to <pattern> to indicate that the pattern is anchored to the beginning (end) of a  
 12074 line of text. Any slash or backslash characters in <pattern> shall be preceded by a backslash  
 12075 character. The anchoring circumflex, dollar sign, and escaping backslash characters shall not be  
 12076 considered part of the search pattern. All other characters in the search pattern shall be  
 12077 considered literal characters.

|       |                                                                                                                           |  |
|-------|---------------------------------------------------------------------------------------------------------------------------|--|
| 12078 | An alternative format is:                                                                                                 |  |
| 12079 | <code>"%s\t%s\t?%s?\n", &lt;identifier&gt;, &lt;filename&gt;, &lt;pattern&gt;</code>                                      |  |
| 12080 | which is identical to the first format except that slashes in <i>&lt;pattern&gt;</i> shall not be preceded by             |  |
| 12081 | escaping backslash characters, and question mark characters in <i>&lt;pattern&gt;</i> shall be preceded by                |  |
| 12082 | backslash characters.                                                                                                     |  |
| 12083 | A second alternative format is:                                                                                           |  |
| 12084 | <code>"%s\t%s\t%d\n", &lt;identifier&gt;, &lt;filename&gt;, &lt;lineno&gt;</code>                                         |  |
| 12085 | where <i>&lt;lineno&gt;</i> is a decimal line number that could be used by an editor to find <i>&lt;identifier&gt;</i> in |  |
| 12086 | <i>&lt;filename&gt;</i> .                                                                                                 |  |
| 12087 | Neither alternative format shall be produced by <i>ctags</i> when it is used as described by                              |  |
| 12088 | IEEE Std. 1003.1-200x, but the standard utilities that process tags files shall be able to process                        |  |
| 12089 | those formats as well as the first format.                                                                                |  |
| 12090 | In any of these formats, the file shall be sorted by identifier, based on the collation sequence in                       |  |
| 12091 | the POSIX locale.                                                                                                         |  |
| 12092 | <b>EXTENDED DESCRIPTION</b>                                                                                               |  |
| 12093 | If the operand identifies C-language source, the <i>ctags</i> utility shall attempt to produce an output                  |  |
| 12094 | line for each of the following objects:                                                                                   |  |
| 12095 | <ul style="list-style-type: none"> <li>• Function definitions</li> </ul>                                                  |  |
| 12096 | <ul style="list-style-type: none"> <li>• Type definitions</li> </ul>                                                      |  |
| 12097 | <ul style="list-style-type: none"> <li>• Macros with arguments</li> </ul>                                                 |  |
| 12098 | It may also produce output for any of the following objects:                                                              |  |
| 12099 | <ul style="list-style-type: none"> <li>• Function prototypes</li> </ul>                                                   |  |
| 12100 | <ul style="list-style-type: none"> <li>• Structures</li> </ul>                                                            |  |
| 12101 | <ul style="list-style-type: none"> <li>• Unions</li> </ul>                                                                |  |
| 12102 | <ul style="list-style-type: none"> <li>• Global variable definitions</li> </ul>                                           |  |
| 12103 | <ul style="list-style-type: none"> <li>• Enumeration types</li> </ul>                                                     |  |
| 12104 | <ul style="list-style-type: none"> <li>• Macros without arguments</li> </ul>                                              |  |
| 12105 | <ul style="list-style-type: none"> <li>• <b>#define</b> statements</li> </ul>                                             |  |
| 12106 | <ul style="list-style-type: none"> <li>• <b>#line</b> statements</li> </ul>                                               |  |
| 12107 | Any <b>#if</b> and <b>#ifdef</b> statements shall produce no output. The tag <b>main</b> is treated specially in C        |  |
| 12108 | programs. The tag formed shall be created by prefixing <b>M</b> to the name of the file, with the                         |  |
| 12109 | trailing <b>.c</b> , and leading path name components (if any) removed.                                                   |  |
| 12110 | On systems that do not support the C-Language Development Utilities option, <i>ctags</i> produces                         |  |
| 12111 | undefined results for C-language source code files.                                                                       |  |
| 12112 | If the operand identifies FORTRAN source, the <i>ctags</i> utility shall produce an output line for each                  |  |
| 12113 | function definition. It may also produce output for any of the following objects:                                         |  |
| 12114 | <ul style="list-style-type: none"> <li>• Subroutine definitions</li> </ul>                                                |  |
| 12115 | <ul style="list-style-type: none"> <li>• COMMON statements</li> </ul>                                                     |  |

- 12116       • PARAMETER statements
- 12117       • DATA and BLOCK DATA statements
- 12118       • Statement numbers
- 12119       On systems that do not support the FORTRAN Development Utilities option, *ctags* produces
- 12120       unspecified results for FORTRAN source code files. It should write to standard error a message
- 12121       identifying this condition and cause a non-zero exit status to be produced.
- 12122       It is implementation-dependent what other objects (including duplicate identifiers) produce
- 12123       output.
- 12124       **EXIT STATUS**
- 12125       The following exit values shall be returned:
- 12126       0   Successful completion.
- 12127       >0   An error occurred.
- 12128       **CONSEQUENCES OF ERRORS**
- 12129       Default.
- 12130       **APPLICATION USAGE**
- 12131       The output with *-x* is meant to be a simple index that can be written out as an off-line readable
- 12132       function index. If the input files to *ctags* (such as *.c* files) were not created using the same locales
- 12133       as those in effect when *ctags -x* is run, results might not be as expected.
- 12134       The description of C-language processing says “attempts to” because the C language can be
- 12135       greatly confused, especially through the use of *#defines*, and this utility would be of no use if
- 12136       the real C preprocessor were run to identify them. The output from *ctags* may be fooled and
- 12137       incorrect for various constructs.
- 12138       Application writers should note that this utility need not be provided on systems that do not
- 12139       support the User Portability Utilities option.
- 12140       **EXAMPLES**
- 12141       None.
- 12142       **RATIONALE**
- 12143       The option list was significantly reduced from that provided by historical implementations. The
- 12144       *-F* option was omitted as redundant, since it is the default. The *-B* option was omitted as being
- 12145       of very limited usefulness. The *-t* option was omitted since the recognition of typedefs is now
- 12146       required for C source files. The *-u* option was omitted because the update function was judged
- 12147       to be not only inefficient, but also rarely needed.
- 12148       An early proposal included a *-w* option to suppress warning diagnostics. Since the types of such
- 12149       diagnostics could not be described, the option was omitted as being not useful.
- 12150       The text for *LC\_CTYPE* about compatibility with the C locale acknowledges that the ISO C
- 12151       standard imposes requirements on the locale used to process C source. This could easily be a
- 12152       superset of that known as “the C locale” by way of implementation extensions, or one of a few
- 12153       alternative locales for systems supporting different codesets. No statement is made for
- 12154       FORTRAN because the ANSI X3.9-1978 standard (FORTRAN 77) does not (yet) define a similar
- 12155       locale concept. However, a general rule in this volume of IEEE Std. 1003.1-200x is that any time
- 12156       that locales do not match (preparing a file for one locale and processing it in another), the results
- 12157       are suspect.
- 12158       The collation sequence of the tags file is not affected by *LC\_COLLATE* because it is typically not
- 12159       used by human readers, but only by programs such as *vi* to locate the tag within the source files.

12160 Using the POSIX locale eliminates some of the problems of coordinating locales between the  
 12161 *ctags* file creator and the *vi* file reader.

12162 Historically, the tags file has been used only by *ex* and *vi*. However, the format of the tags file  
 12163 has been published to encourage other programs to use the tags in new ways. The format allows  
 12164 either BREs or line numbers to find the identifiers because the historical *vi* recognizes either. The  
 12165 *ctags* utility does not produce the format using line numbers because it is not useful following  
 12166 any source file changes that add or delete lines. The documented search patterns match  
 12167 historical practice. It should be noted that literal leading circumflex or trailing dollar-sign  
 12168 characters in the search pattern will only behave correctly if anchored to the beginning of the  
 12169 line or end of the line by an additional circumflex or dollar-sign character.

12170 Historical implementations also understand the objects used by the languages Pascal and  
 12171 sometimes LISP, and they understand the C source output by *lex* and *yacc*. The *ctags* utility is  
 12172 not required to accommodate these languages, although implementors are encouraged to do so.

12173 The following historical option was not specified, as *vgrind* is not included in this volume of  
 12174 IEEE Std. 1003.1-200x:

12175        -v           If the -v flag is given, an index of the form expected by *vgrind* is produced on the  
 12176                      standard output. This listing contains the function name, file name, and page  
 12177                      number (assuming 64-line pages). Since the output is sorted into lexicographic  
 12178                      order, it may be desired to run the output through *sort -f*. Sample use:

12179                      ctags -v files | sort -f > index vgrind -x index

12180 The special treatment of the tag **main** makes the use of *ctags* practical in directories with more  
 12181 than one program.

#### 12182 FUTURE DIRECTIONS

12183        None.

#### 12184 SEE ALSO

12185        *c89*, *fort77*, *vi*

#### 12186 CHANGE HISTORY

12187        First released in Issue 4.

#### 12188 Issue 5

12189        FUTURE DIRECTIONS section added.

#### 12190 Issue 6

12191        This utility is now marked as part of the User Portability Utilities option.

12192        The OUTPUT FILES section is changed to align with the IEEE P1003.2b draft standard.

12193        The normative text is reworded to avoid use of the term “must” for application requirements.

12194 **NAME**

12195           cut — cut out selected fields of each line of a file

12196 **SYNOPSIS**12197           cut -b *list* [-n] [*file* ...]12198           cut -c *list* [*file* ...]12199           cut -f *list* [-d *delim*][-s][*file* ...]12200 **DESCRIPTION**

12201           The *cut* utility shall cut out bytes (**-b** option), characters (**-c** option) or character-delimited fields  
 12202           (**-f** option) from each line in one or more files, concatenate them, and write them to standard  
 12203           output.

12204 **OPTIONS**

12205           The *cut* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 12206           Section 12.2, Utility Syntax Guidelines.

12207           The application shall ensure that the option-argument *list* (see options **-b**, **-c**, and **-f** below) is a  
 12208           comma-separated list or <blank> character-separated list of positive numbers and ranges.  
 12209           Ranges can be in three forms. The first is two positive numbers separated by a hyphen  
 12210           (*low-high*), which represents all fields from the first number to the second number. The second is  
 12211           a positive number preceded by a hyphen (*-high*), which represents all fields from field number 1  
 12212           to that number. The third is a positive number followed by a hyphen (*low-*), which represents  
 12213           that number to the last field, inclusive. The elements in *list* can be repeated, can overlap, and can  
 12214           be specified in any order, but the bytes, characters, or fields selected shall be written in the order  
 12215           of the input data. If an element appears in the selection list more than once, it shall be written  
 12216           exactly once.

12217           The following options shall be supported:

12218           **-b list**       Cut based on a *list* of bytes. Each selected byte shall be output unless the **-n** option  
 12219           is also specified. It shall not be an error to select bytes not present in the input line.

12220           **-c list**       Cut based on a *list* of characters. Each selected character shall be output. It shall  
 12221           not be an error to select characters not present in the input line.

12222           **-d delim**     Set the field delimiter to the character *delim*. The default is the <tab> character.

12223           **-f list**       Cut based on a *list* of fields, assumed to be separated in the file by a delimiter  
 12224           character (see **-d**). Each selected field shall be output. Output fields shall be  
 12225           separated by a single occurrence of the field delimiter character. Lines with no field  
 12226           delimiters shall be passed through intact, unless **-s** is specified. It shall not be an  
 12227           error to select fields not present in the input line.

12228           **-n**           Do not split characters. When specified with the **-b** option, each element in *list* of  
 12229           the form *low-high* (hyphen-separated numbers) shall be modified as follows:

- 12230           • If the byte selected by *low* is not the first byte of a character, *low* shall be  
 12231           decremented to select the first byte of the character originally selected by *low*.  
 12232           If the byte selected by *high* is not the last byte of a character, *high* shall be  
 12233           decremented to select the last byte of the character prior to the character  
 12234           originally selected by *high*, or zero if there is no prior character. If the resulting  
 12235           range element has *high* equal to zero or *low* greater than *high*, the list element  
 12236           shall be dropped from *list* for that input line without causing an error.

12237           Each element in *list* of the form *low-* shall be treated as above with *high* set to the  
 12238           number of bytes in the current line, not including the terminating <newline>

|       |                              |                                                                                                                     |
|-------|------------------------------|---------------------------------------------------------------------------------------------------------------------|
| 12239 |                              | character. Each element in <i>list</i> of the form <i>–high</i> shall be treated as above with <i>low</i>           |
| 12240 |                              | set to 1. Each element in <i>list</i> of the form <i>num</i> (a single number) shall be treated as                  |
| 12241 |                              | above with <i>low</i> set to <i>num</i> and <i>high</i> set to <i>num</i> .                                         |
| 12242 | <b>–s</b>                    | Suppress lines with no delimiter characters, when used with the <b>–f</b> option. Unless                            |
| 12243 |                              | specified, lines with no delimiters shall be passed through untouched.                                              |
| 12244 | <b>OPERANDS</b>              |                                                                                                                     |
| 12245 |                              | The following operand shall be supported:                                                                           |
| 12246 | <i>file</i>                  | A path name of an input file. If no <i>file</i> operands are specified, or if a <i>file</i> operand is              |
| 12247 |                              | '–', the standard input shall be used.                                                                              |
| 12248 | <b>STDIN</b>                 |                                                                                                                     |
| 12249 |                              | The standard input shall be used only if no <i>file</i> operands are specified, or if a <i>file</i> operand is '–'. |
| 12250 |                              | See the INPUT FILES section.                                                                                        |
| 12251 | <b>INPUT FILES</b>           |                                                                                                                     |
| 12252 |                              | The input files shall be text files, except that line lengths shall be unlimited.                                   |
| 12253 | <b>ENVIRONMENT VARIABLES</b> |                                                                                                                     |
| 12254 |                              | The following environment variables shall affect the execution of <i>cut</i> :                                      |
| 12255 | <i>LANG</i>                  | Provide a default value for the internationalization variables that are unset or null.                              |
| 12256 |                              | If <i>LANG</i> is unset or null, the corresponding value from the implementation-                                   |
| 12257 |                              | dependent default locale shall be used. If any of the internationalization variables                                |
| 12258 |                              | contains an invalid setting, the utility shall behave as if none of the variables had                               |
| 12259 |                              | been defined.                                                                                                       |
| 12260 | <i>LC_ALL</i>                | If set to a non-empty string value, override the values of all the other                                            |
| 12261 |                              | internationalization variables.                                                                                     |
| 12262 | <i>LC_CTYPE</i>              | Determine the locale for the interpretation of sequences of bytes of text data as                                   |
| 12263 |                              | characters (for example, single-byte as opposed to multi-byte characters in                                         |
| 12264 |                              | arguments and input files).                                                                                         |
| 12265 | <i>LC_MESSAGES</i>           |                                                                                                                     |
| 12266 |                              | Determine the locale that should be used to affect the format and contents of                                       |
| 12267 |                              | diagnostic messages written to standard error.                                                                      |
| 12268 | XSI <i>NLSPATH</i>           | Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .                               |
| 12269 | <b>ASYNCHRONOUS EVENTS</b>   |                                                                                                                     |
| 12270 |                              | Default.                                                                                                            |
| 12271 | <b>STDOUT</b>                |                                                                                                                     |
| 12272 |                              | The <i>cut</i> utility output shall be a concatenation of the selected bytes, characters, or fields (one of         |
| 12273 |                              | the following):                                                                                                     |
| 12274 |                              | "%s\n", <concatenation of bytes>                                                                                    |
| 12275 |                              | "%s\n", <concatenation of characters>                                                                               |
| 12276 |                              | "%s\n", <concatenation of fields and field delimiters>                                                              |
| 12277 | <b>STDERR</b>                |                                                                                                                     |
| 12278 |                              | Used only for diagnostic messages.                                                                                  |



12279 **OUTPUT FILES**

12280 None.

12281 **EXTENDED DESCRIPTION**

12282 None.

12283 **EXIT STATUS**

12284 The following exit values shall be returned:

12285 0 All input files were output successfully.

12286 &gt;0 An error occurred.

12287 **CONSEQUENCES OF ERRORS**

12288 Default.

12289 **APPLICATION USAGE**

12290 Earlier versions of the *cut* utility worked in an environment where bytes and characters were  
 12291 considered equivalent (modulo <backspace> and <tab> character processing in some  
 12292 implementations). In the extended world of multi-byte characters, the new **-b** option has been  
 12293 added. The **-n** option (used with **-b**) allows it to be used to act on bytes rounded to character  
 12294 boundaries. The algorithm specified for **-n** guarantees that:

12295 `cut -b 1-500 -n file > file1`12296 `cut -b 501- -n file > file2`

12297 ends up with all the characters in **file** appearing exactly once in **file1** or **file2**. (There is,  
 12298 however, a <newline> character in both **file1** and **file2** for each <newline> character in **file**.)

12299 **EXAMPLES**

12300 Examples of the option qualifier list:

12301 1,4,7 Select the first, fourth, and seventh bytes, characters, or fields and field delimiters.

12302 1-3,8 Equivalent to 1,2,3,8.

12303 -5,10 Equivalent to 1,2,3,4,5,10.

12304 3- Equivalent to third to last, inclusive.

12305 The *low-high* forms are not always equivalent when used with **-b** and **-n** and multi-byte  
 12306 characters; see the description of **-n**.

12307 The following command:

12308 `cut -d : -f 1,6 /etc/passwd`

12309 reads the System V password file (user database) and produces lines of the form:

12310 `<user ID>:<home directory>`

12311 Most utilities in this volume of IEEE Std. 1003.1-200x work on text files. The *cut* utility can be  
 12312 used to turn files with arbitrary line lengths into a set of text files containing the same data. The  
 12313 *paste* utility can be used to create (or recreate) files with arbitrary line lengths. For example, if **file**  
 12314 contains long lines:

12315 `cut -b 1-500 -n file > file1`12316 `cut -b 501- -n file > file2`

12317 creates **file1** (a text file) with lines no longer than 500 bytes (plus the <newline> character) and  
 12318 **file2** that contains the remainder of the data from **file**. (Note that **file2** is not a text file if there  
 12319 are lines in **file** that are longer than 500 + {LINE\_MAX} bytes.) The original file can be recreated  
 12320 from **file1** and **file2** using the command:

12321        paste -d "\0" file1 file2 > file

## 12322 RATIONALE

12323        Some historical implementations do not count <backspace> characters in determining character  
12324        counts with the `-c` option. This may be useful for using *cut* for processing *nroff* output. It was  
12325        deliberately decided not to have the `-c` option treat either <backspace> or <tab> characters in  
12326        any special fashion. The *fold* utility does treat these characters specially.

12327        Unlike other utilities, some historical implementations of *cut* exit after not finding an input file,  
12328        rather than continuing to process the remaining *file* operands. This behavior is prohibited by this  
12329        volume of IEEE Std. 1003.1-200x, where only the exit status is affected by this problem.

12330        The behavior of *cut* when provided with either mutually-exclusive options or options that do  
12331        not work logically together has been deliberately left unspecified in favor of global wording in  
12332        Section 1.11 on page 25.

12333        The OPTIONS section was changed in response to P1003.2-N149. The change represents  
12334        historical practice on all known systems. The original standard was ambiguous on the nature of  
12335        the output.

12336        The *list* option-arguments are historically used to select the portions of the line to be written, but  
12337        do not affect the order of the data. For example:

12338        echo abcdefghi | cut -c6,2,4-7,1

12339        yields "abdefg".

12340        A proposal to enhance *cut* with the following option:

12341        `-o` Preserve the selected field order. When this option is specified, each byte, character, or field  
12342        (or ranges of such) shall be written in the order specified by the *list* option-argument, even if  
12343        this requires multiple outputs of the same bytes, characters, or fields.

12344        was rejected because this type of enhancement is outside the scope of the IEEE P1003.2b draft  
12345        standard.

## 12346 FUTURE DIRECTIONS

12347        None.

## 12348 SEE ALSO

12349        *grep*, *paste*, Section 2.5 on page 43

## 12350 CHANGE HISTORY

12351        First released in Issue 2.

## 12352 Issue 4

12353        Aligned with the ISO/IEC 9945-2:1993 standard.

## 12354 Issue 6

12355        The OPTIONS section is changed to align with the IEEE P1003.2b draft standard.

12356        The normative text is reworded to avoid use of the term “must” for application requirements.

12357 **NAME**12358       cxref — generate a C-language program cross-reference table (**DEVELOPMENT**)12359 **SYNOPSIS**

```
12360 xsi cxref [-cs][-o file][-w num] [-D name[=def]]... [-I dir]...
12361 [-U name]... file ...
```

12362

12363 **DESCRIPTION**

12364       The *cxref* utility shall analyze a collection of C-language *files* and attempt to build a cross-  
 12365       reference table. Information from **#define** lines is included in the symbol table. A sorted listing  
 12366       shall be written to standard output of all symbols (auto, static, and global) in each *file* separately,  
 12367       or with the **-c** option, in combination. Each symbol contains an asterisk before the declaring  
 12368       reference.

12369 **OPTIONS**

12370       The *cxref* utility shall conform to the System Interface Definitions volume of  
 12371       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines, except that the order of the **-D**, **-I**,  
 12372       and **-U** options (which are identical to their interpretation by *c89*) is significant. The following  
 12373       options shall be supported:

12374       **-c**           Write a combined cross-reference of all input files.

12375       **-w num**       Format output no wider than *num* (decimal) columns. This option defaults to 80 if  
 12376       *num* is not specified or is less than 51.

12377       **-o file**       Direct output to named *file*.

12378       **-s**           Operate silently; do not print input file names.

12379 **OPERANDS**

12380       The following operand shall be supported:

12381       *file*           A path name of a C-language source file.

12382 **STDIN**

12383       Not used.

12384 **INPUT FILES**

12385       The input files are C-language source files.

12386 **ENVIRONMENT VARIABLES**

12387       The following environment variables shall affect the execution of *cxref*:

12388       **LANG**       Provide a default value for the internationalization variables that are unset or null.  
 12389       If *LANG* is unset or null, the corresponding value from the implementation-  
 12390       dependent default locale shall be used. If any of the internationalization variables  
 12391       contains an invalid setting, the utility shall behave as if none of the variables had  
 12392       been defined.

12393       **LC\_ALL**       If set to a non-empty string value, override the values of all the other  
 12394       internationalization variables.

12395       **LC\_COLLATE**

12396       Determine the locale for the ordering of the output.

12397       **LC\_CTYPE**   Determine the locale for the interpretation of sequences of bytes of text data as  
 12398       characters (for example, single-byte as opposed to multi-byte characters in  
 12399       arguments and input files).

- 12400 **LC\_MESSAGES**  
12401 Determine the locale that should be used to affect the format and contents of  
12402 diagnostic messages written to standard error.
- 12403 **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.
- 12404 **ASYNCHRONOUS EVENTS**  
12405 Default.
- 12406 **STDOUT**  
12407 The standard output shall be used for the cross-reference listing, unless the **-o** option is used to  
12408 select a different output file.
- 12409 The format of standard output is unspecified, except that the following information shall be  
12410 included:
- 12411 • If the **-c** option is not specified, each portion of the listing starts with the name of the input  
12412 file on a separate line.
  - 12413 • The name line is followed by a sorted list of symbols, each with its associated location path  
12414 name, the name of the function in which it appears (if it is not a function name itself), and  
12415 line number references.
  - 12416 • Each line number may be preceded by an asterisk (**'\*'**) flag, meaning that this is the  
12417 declaring reference. Other single-character flags, with implementation-dependent meanings,  
12418 may be included.
- 12419 **STDERR**  
12420 Used only for diagnostic messages.
- 12421 **OUTPUT FILES**  
12422 The output file named by the **-o** option shall be used instead of standard output.
- 12423 **EXTENDED DESCRIPTION**  
12424 None.
- 12425 **EXIT STATUS**  
12426 The following exit values shall be returned:
- 12427 0 Successful completion.
  - 12428 >0 An error occurred.
- 12429 **CONSEQUENCES OF ERRORS**  
12430 Default.
- 12431 **APPLICATION USAGE**  
12432 Application writers should note that this utility need not be provided on systems that do not  
12433 support the XSI Development Utilities option.
- 12434 **EXAMPLES**  
12435 None.
- 12436 **RATIONALE**  
12437 None.
- 12438 **FUTURE DIRECTIONS**  
12439 None.

12440 **SEE ALSO**

12441 *c89*

12442 **CHANGE HISTORY**

12443 First released in Issue 2.

12444 **Issue 4**

12445 Format reorganized.

12446 Utility Syntax Guidelines support mandated.

12447 Internationalized environment variable support mandated.

12448 **Issue 5**

12449 In the SYNOPSIS, [-U *dir*]*ischangedto*[-U *name*].

12450 **Issue 6**

12451 The APPLICATION USAGE section is added.

12452 **NAME**

12453           date — write the date and time

12454 **SYNOPSIS**

12455           date [-u] [+format]

12456 XSI       date [-u] mmddhhmm[[cc]yy]

12457

12458 **DESCRIPTION**

12459 XSI       The *date* utility shall write the date and time to standard output or attempt to set the system date  
 12460 and time. By default, the current date and time shall be written. If an operand beginning with  
 12461 '+' is specified, the output format of *date* shall be controlled by the field descriptors and other  
 12462 text in the operand.

12463 **OPTIONS**

12464       The *date* utility shall conform to the System Interface Definitions volume of  
 12465 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

12466       The following option shall be supported:

12467       -u       Perform operations as if the *TZ* environment variable was set to the string "UTC0",  
 12468               or its equivalent historical value of "GMT0". Otherwise, *date* shall use the  
 12469               timezone indicated by the *TZ* environment variable or the system default if that  
 12470               variable is not set.

12471 **OPERANDS**

12472       The following operands shall be supported:

12473       +format   When the format is specified, each field descriptor shall be replaced in the  
 12474               standard output by its corresponding value. All other characters shall be copied to  
 12475               the output without change. The output always shall be terminated with a  
 12476               <newline> character.

12477 **Notes to Reviewers**12478       *This section with side shading will not appear in the final copy. - Ed.*

12479       D1, XCU, ERN 195 notes that there are differences between the field descriptors  
 12480       and the c9X definition for *strftime()*. This is included on the issues list.

12481 **Field Descriptors**

12482       %a       Locale's abbreviated weekday name.  
 12483       %A       Locale's full weekday name.  
 12484       %b       Locale's abbreviated month name.  
 12485       %B       Locale's full month name.  
 12486       %c       Locale's appropriate date and time representation.  
 12487       %C       Century (a year divided by 100 and truncated to an integer) as a decimal  
 12488               number [00-99].  
 12489       %d       Day of the month as a decimal number [01-31].  
 12490       %D       Date in the format *mm/dd/yy*.  
 12491       %e       Day of the month as a decimal number [1-31] in a two-digit field with  
 12492               leading space character fill.

|       |    |                                                                                      |
|-------|----|--------------------------------------------------------------------------------------|
| 12493 | %h | A synonym for %b.                                                                    |
| 12494 | %H | Hour (24-hour clock) as a decimal number [00-23].                                    |
| 12495 | %I | Hour (12-hour clock) as a decimal number [01-12].                                    |
| 12496 | %j | Day of the year as a decimal number [001-366].                                       |
| 12497 | %m | Month as a decimal number [01-12].                                                   |
| 12498 | %M | Minute as a decimal number [00-59].                                                  |
| 12499 | %n | A <newline> character.                                                               |
| 12500 | %p | Locale's equivalent of either AM or PM.                                              |
| 12501 | %r | 12-hour clock time [01-12] using the AM/PM notation; in the POSIX                    |
| 12502 |    | locale, this is equivalent to %I:%M:%S% p.                                           |
| 12503 | %S | Seconds as a decimal number [00-61].                                                 |
| 12504 | %t | A <tab> character.                                                                   |
| 12505 | %T | 24-hour clock time [00-23] in the format HH:MM:SS.                                   |
| 12506 | %u | Weekday as a decimal number [1 (Monday)-7].                                          |
| 12507 | %U | Week of the year (Sunday as the first day of the week) as a decimal                  |
| 12508 |    | number [00-53]. All days in a new year preceding the first Sunday shall be           |
| 12509 |    | considered to be in week 0.                                                          |
| 12510 | %V | Week of the year (Monday as the first day of the week) as a decimal                  |
| 12511 |    | number [01-53]. If the week containing January 1 has four or more days in            |
| 12512 |    | the new year, then it shall be considered week 1; otherwise, it shall be the         |
| 12513 |    | last week of the previous year, and the next week shall be week 1.                   |
| 12514 | %w | Weekday as a decimal number [0 (Sunday)-6].                                          |
| 12515 | %W | Week of the year (Monday as the first day of the week) as a decimal                  |
| 12516 |    | number [00-53]. All days in a new year preceding the first Monday shall              |
| 12517 |    | be considered to be in week 0.                                                       |
| 12518 | %x | Locale's appropriate date representation.                                            |
| 12519 | %X | Locale's appropriate time representation.                                            |
| 12520 | %y | Year within century [00-99].                                                         |
| 12521 | %Y | Year with century as a decimal number.                                               |
| 12522 | %Z | Timezone name, or no characters if no timezone is determinable.                      |
| 12523 | %% | A percent sign character.                                                            |
| 12524 |    | See the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 7.3.5, |
| 12525 |    | LC_TIME for the field descriptor values in the POSIX locale.                         |

|           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                   |
|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| 12526     | <b>Modified Field Descriptors</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |                                                                                                                                                   |
| 12527     | Some field descriptors can be modified by the <i>E</i> and <i>O</i> modifier characters to indicate a different format or specification as specified in the <i>LC_TIME</i> locale description (see the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 7.3.5, <i>LC_TIME</i> ). If the corresponding keyword (see <b>era</b> , <b>era_year</b> , <b>era_d_fmt</b> , and <b>alt_digits</b> in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 7.3.5, <i>LC_TIME</i> ) is not specified or not supported for the current locale, the unmodified field descriptor value shall be used.                                                                                         |                                                                                                                                                   |
| 12528     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                   |
| 12529     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                   |
| 12530     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                   |
| 12531     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                   |
| 12532     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                   |
| 12533     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                   |
| 12534     | <b>%Ec</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Locale's alternative appropriate date and time representation.                                                                                    |
| 12535     | <b>%EC</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | The name of the base year (period) in the locale's alternative representation.                                                                    |
| 12536     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                   |
| 12537     | <b>%Ex</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Locale's alternative date representation.                                                                                                         |
| 12538     | <b>%EX</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Locale's alternative time representation.                                                                                                         |
| 12539     | <b>%Ey</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Offset from <b>%EC</b> (year only) in the locale's alternative representation.                                                                    |
| 12540     | <b>%EY</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Full alternative year representation.                                                                                                             |
| 12541     | <b>%Od</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Day of month using the locale's alternative numeric symbols.                                                                                      |
| 12542     | <b>%Oe</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Day of month using the locale's alternative numeric symbols.                                                                                      |
| 12543     | <b>%OH</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Hour (24-hour clock) using the locale's alternative numeric symbols.                                                                              |
| 12544     | <b>%OI</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Hour (12-hour clock) using the locale's alternative numeric symbols.                                                                              |
| 12545     | <b>%Om</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Month using the locale's alternative numeric symbols.                                                                                             |
| 12546     | <b>%OM</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Minutes using the locale's alternative numeric symbols.                                                                                           |
| 12547     | <b>%OS</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Seconds using the locale's alternative numeric symbols.                                                                                           |
| 12548     | <b>%Ou</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Weekday as a number in the locale's alternative representation (Monday = 1).                                                                      |
| 12549     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                   |
| 12550     | <b>%OU</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Week number of the year (Sunday as the first day of the week) using the locale's alternative numeric symbols.                                     |
| 12551     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                   |
| 12552     | <b>%OV</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Week number of the year (Monday as the first day of the week, rules corresponding to <b>%V</b> ), using the locale's alternative numeric symbols. |
| 12553     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                   |
| 12554     | <b>%Ow</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Weekday as a number in the locale's alternative representation (Sunday = 0).                                                                      |
| 12555     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                   |
| 12556     | <b>%OW</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Week number of the year (Monday as the first day of the week) using the locale's alternative numeric symbols.                                     |
| 12557     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                   |
| 12558     | <b>%Oy</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Year (offset from <b>%C</b> ) in alternative representation.                                                                                      |
| 12559 XSI | <b>mmddhhmm[[cc]yy]</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |                                                                                                                                                   |
| 12560     | Attempt to set the system date and time from the value given in the operand. This is only possible if the user has appropriate privileges and the system permits the setting of the system date and time. The first <i>mm</i> is the month (number); <i>dd</i> is the day (number); <i>hh</i> is the hour (number, 24-hour system); the second <i>mm</i> is the minute (number); <i>cc</i> is the century and is the first two digits of the year (this is optional); <i>yy</i> is the last two digits of the year and is optional. If century is not specified, then values in the range [69-99] shall refer to years 1969 to 1999 inclusive, and values in the range [00-68] shall refer to years 2000 to 2068 inclusive. |                                                                                                                                                   |
| 12561     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                   |
| 12562     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                   |
| 12563     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                   |
| 12564     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                   |
| 12565     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                   |
| 12566     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                   |
| 12567     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |                                                                                                                                                   |



12568 **STDIN**

12569 Not used.

12570 **INPUT FILES**

12571 None.

12572 **ENVIRONMENT VARIABLES**12573 The following environment variables shall affect the execution of *date*:

12574 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 12575 If *LANG* is unset or null, the corresponding value from the implementation-  
 12576 dependent default locale shall be used. If any of the internationalization variables  
 12577 contains an invalid setting, the utility shall behave as if none of the variables had  
 12578 been defined.

12579 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 12580 internationalization variables.

12581 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 12582 characters (for example, single-byte as opposed to multi-byte characters in  
 12583 arguments).

12584 *LC\_MESSAGES*  
 12585 Determine the locale that should be used to affect the format and contents of  
 12586 diagnostic messages written to standard error.

12587 *LC\_TIME* Determine the format and contents of date and time strings written by *date*.

12588 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

12589 *TZ* Determine the timezone in which the time and date are written, unless the *-u*  
 12590 option is specified. If the *TZ* variable is not set and the *-u* is not specified, an  
 12591 unspecified system default timezone is used.

12592 **ASYNCHRONOUS EVENTS**

12593 Default.

12594 **STDOUT**

12595 When no formatting operand is specified, the output in the POSIX locale shall be equivalent to  
 12596 specifying:

12597 *date* "+%a %b %e %H:%M:%S %Z %Y"

12598 **STDERR**

12599 Used only for diagnostic messages.

12600 **OUTPUT FILES**

12601 None.

12602 **EXTENDED DESCRIPTION**

12603 None.

12604 **EXIT STATUS**

12605 The following exit values shall be returned:

12606 0 The date was written successfully.

12607 >0 An error occurred.

## 12608 CONSEQUENCES OF ERRORS

12609 Default.

## 12610 APPLICATION USAGE

12611 Field descriptors are of unspecified format when not in the POSIX locale. Some of them can  
 12612 contain <newline> characters in some locales, so it may be difficult to use the format shown in  
 12613 standard output for parsing the output of *date* in those locales.

12614 The range of values for %S extends from 0 to 61 seconds to accommodate the occasional leap  
 12615 second or double leap second.

12616 Although certain of the field descriptors in the POSIX locale (such as the name of the month) are  
 12617 shown with initial capital letters, this need not be the case in other locales. Programs using these  
 12618 fields may need to adjust the capitalization if the output is going to be used at the beginning of a  
 12619 sentence.

12620 The date string formatting capabilities are intended for use in Gregorian-style calendars,  
 12621 possibly with a different starting year (or years). The %x and %c field descriptors, however, are  
 12622 intended for local representation; these may be based on a different, non-Gregorian calendar.

12623 The %C field descriptor was introduced to allow a fallback for the %EC (alternative year format  
 12624 base year); it can be viewed as the base of the current subdivision in the Gregorian calendar. A  
 12625 century is not calculated as an ordinal number; this Guide was published in century 19, not the  
 12626 twentieth. Both the %Ey and %y can then be viewed as the offset from %EC and %C,  
 12627 respectively.

12628 The *E* and *O* modifiers modify the traditional field descriptors, so that they can always be used,  
 12629 even if the implementation (or the current locale) does not support the modifier.

12630 The *E* modifier supports alternative date formats, such as the Japanese Emperor's Era, as long as  
 12631 these are based on the Gregorian calendar system. Extending the *E* modifiers to other date  
 12632 elements may provide an implementation-dependent extension capable of supporting other  
 12633 calendar systems, especially in combination with the *O* modifier.

12634 The *O* modifier supports time and date formats using the locale's alternative numerical symbols,  
 12635 such as Kanji or Hindi digits or ordinal number representation.

12636 Non-European locales, whether they use Latin digits in computational items or not, often have  
 12637 local forms of the digits for use in date formats. This is not totally unknown even in Europe; a  
 12638 variant of dates uses Roman numerals for the months: the third day of September 1991 would be  
 12639 written as 3.IX.1991. In Japan, Kanji digits are regularly used for dates; in Arabic-speaking  
 12640 countries, Hindi digits are used. The %d, %e, %H, %I, %m, %S, %U, %w, %W, and %y field  
 12641 descriptors always return the date and time field in Latin digits (that is, 0 to 9). The %O modifier  
 12642 was introduced to support the use for display purposes of non-Latin digits. In the *LC\_TIME*  
 12643 category in *localedef*, the optional **alt\_digits** keyword is intended for this purpose. As an  
 12644 example, assume the following (partial) *localedef* source:

```
12645 alt_digits "";"I";"II";"III";"IV";"V";"VI";"VII";"VIII" \
12646 "IX";"X";"XI";"XII"
12647 d_fmt "%e.%Om.%Y"
```

12648 With the above date, the command:

```
12649 date "+%x"
```

12650 would yield 3.IX.1991. With the same **d\_fmt**, but without the **alt\_digits**, the command would  
 12651 yield 3.9.1991.

12652 **EXAMPLES**

- 12653           1. The following are input/output examples of *date* used at arbitrary times in the POSIX  
12654 locale:
- 12655           \$ date  
12656           **Tue Jun 26 09:58:10 PDT 1990**
- 12657           \$ date "+DATE: %m/%d/%Y%nTIME: %H:%M:%S"  
12658           **DATE: 11/02/91**  
12659           **TIME: 13:36:16**
- 12660           \$ date "+TIME: %r"  
12661           **TIME: 01:36:32 PM**
- 12662           2. Examples for Denmark, where the default date and time format is *%a %d %b %Y %T %Z*:
- 12663           \$ LANG=da\_DK.iso\_8859-1 date  
12664           **ons 02 okt 1991 15:03:32 CET**
- 12665           \$ LANG=da\_DK.iso\_8859-1 date "+DATO: %A den %e. %B nKLOKKEN: %H:%M:%S"  
12666           **DATO: onsdag den 2. oktober 1991**  
12667           **KLOKKEN: 15:03:56**
- 12668           3. Examples for Germany, where the default date and time format is *%a %d.%h.%Y, %T %Z*:
- 12669           \$ LANG=De\_DE.88591 date  
12670           **Mi 02.Okt.1991, 15:01:21 MEZ**
- 12671           \$ LANG=De\_DE.88591 date "+DATUM: %A, %d. %B nZEIT: %H:%M:%S"  
12672           **DATUM: Mittwoch, 02. Oktober 1991**  
12673           **ZEIT: 15:02:02**
- 12674           4. Examples for France, where the default date and time format is *%a %d %h %Y %Z %T*:
- 12675           \$ LANG=Fr\_FR.88591 date  
12676           **Mer 02 oct 1991 MET 15:03:32**
- 12677           \$ LANG=Fr\_FR.88591 date "+JOUR: %A %d %B nHEURE: %H:%M:%S"  
12678           **JOUR: Mercredi 02 octobre 1991**  
12679           **HEURE: 15:03:56**

12680 **RATIONALE**

- 12681           Some of the new options for formatting are from the ISO C standard. The *-u* option was  
12682 introduced to allow portable access to Coordinated Universal Time (UTC). The string "GMT0" is  
12683 allowed as an equivalent TZ value to be compatible with all of the systems using the BSD  
12684 implementation, where this option originated.
- 12685           The *%e* format field descriptor (adopted from System V) was added because the ISO C standard  
12686 descriptors did not provide any way to produce the historical default *date* output during the first  
12687 nine days of any month.
- 12688           There are two varieties of day and week numbering supported (in addition to any others created  
12689 with the locale-dependent *%E* and *%O* modifier characters):
- 12690           • The historical variety in which Sunday is the first day of the week and the weekdays  
12691           preceding the first Sunday of the year are considered week 0. These are represented by *%w*  
12692           and *%U*. A variant of this is *%W*, using Monday as the first day of the week, but still referring  
12693           to week 0. This view of the calendar was retained because so many historical applications  
12694           depend on it and the ISO C standard *strftime()* function, on which many *date*  
12695           implementations are based, was defined in this way.

- 12696       • The international standard, based on the ISO 8601:1988 standard where Monday is the first  
12697       weekday and the algorithm for the first week number is more complex: If the week (Monday  
12698       to Sunday) containing January 1 has four or more days in the new year, then it is week 1;  
12699       otherwise, it is week 53 of the previous year, and the next week is week 1. These are  
12700       represented by the new field descriptors %u and %V, added as a result of international  
12701       comments.
- 12702       The %C field descriptor was introduced to allow a fallback for the %EC (alternate year format  
12703       base year); it can be viewed as the base of the current subdivision in the Gregorian calendar. A  
12704       century is not calculated as an ordinal number. The original version of this volume of  
12705       IEEE Std. 1003.1-200x was approved in century 19, not the twentieth. Both the %Ey and %y can  
12706       then be viewed as the offset from %EC and %C, respectively.
- 12707 **FUTURE DIRECTIONS**
- 12708       None.
- 12709 **SEE ALSO**
- 12710       The System Interfaces volume of IEEE Std. 1003.1-200x, *ctime()*, *printf()*
- 12711 **CHANGE HISTORY**
- 12712       First released in Issue 2.
- 12713 **Issue 4**
- 12714       Aligned with the ISO/IEC 9945-2:1993 standard.
- 12715 **Issue 5**
- 12716       Changes are made for Year 2000 alignment.
- 12717 **Issue 6**
- 12718       The following new requirements on POSIX implementations derive from alignment with the  
12719       Single UNIX Specification:
- 12720       • The setting of system date and time is described, including how to interpret two-digit year  
12721       values if a century is not given.
- 12722       • The %EX modified field descriptor is added.

## 12723 NAME

12724 dd — convert and copy a file

## 12725 SYNOPSIS

12726 dd [*operand* ...]

## 12727 DESCRIPTION

12728 The *dd* utility shall copy the specified input file to the specified output file with possible  
 12729 conversions using specific input and output block sizes. It shall read the input one block at a  
 12730 time, using the specified input block size; it shall then process the block of data actually  
 12731 returned, which could be smaller than the requested block size. It shall apply any conversions  
 12732 that have been specified and write the resulting data to the output in blocks of the specified  
 12733 output block size. If the **bs=expr** operand is specified and no conversions other than **sync**,  
 12734 **noerror**, or **notrunc** are requested, the data returned from each input block shall be written as a  
 12735 separate output block; if the read returns less than a full block and the **sync** conversion is not  
 12736 specified, the resulting output block shall be the same size as the input block. If the **bs=expr**  
 12737 operand is not specified, or a conversion other than **sync**, **noerror**, or **notrunc** is requested, the  
 12738 input shall be processed and collected into full-sized output blocks until the end of the input is  
 12739 reached.

12740 The processing order shall be as follows:

- 12741 1. An input block is read.
- 12742 2. If the input block is shorter than the specified input block size and the **sync** conversion is  
 12743 specified, null bytes shall be appended to the input data up to the specified size. (If either  
 12744 **block** or **unblock** is also specified, <space> characters shall be appended instead of null  
 12745 bytes.) The remaining conversions and output shall include the pad characters as if they  
 12746 had been read from the input.
- 12747 3. If the **bs=expr** operand is specified and no conversion other than **sync** or **noerror** is  
 12748 requested, the resulting data shall be written to the output as a single block, and the  
 12749 remaining steps are omitted.
- 12750 4. If the **swab** conversion is specified, each pair of input data bytes shall be swapped. If there  
 12751 is an odd number of bytes in the input block, the last byte in the input record shall not be  
 12752 swapped.
- 12753 5. Any remaining conversions (**block**, **unblock**, **lcase**, and **ucase**) shall be performed. These  
 12754 conversions shall operate on the input data independently of the input blocking; an input  
 12755 or output fixed-length record may span block boundaries.
- 12756 6. The data resulting from input or conversion or both shall be aggregated into output blocks  
 12757 of the specified size. After the end of input is reached, any remaining output shall be  
 12758 written as a block without padding if **conv=sync** is not specified; thus, the final output  
 12759 block may be shorter than the output block size.

## 12760 OPTIONS

12761 None.

## 12762 OPERANDS

12763 All of the operands shall be processed before any input is read. The following operands shall be  
 12764 supported:

- |       |                |                                                                                        |
|-------|----------------|----------------------------------------------------------------------------------------|
| 12765 | <b>if=file</b> | Specify the input path name; the default is standard input.                            |
| 12766 | <b>of=file</b> | Specify the output path name; the default is standard output. If the <b>seek=expr</b>  |
| 12767 |                | conversion is not also specified, the output file shall be truncated before the copy   |
| 12768 |                | begins, unless <b>conv=notrunc</b> is specified. If <b>seek=expr</b> is specified, but |

|           |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----------|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 12769     |                               | <b>conv=notrunc</b> is not, the effect of the copy shall be to preserve the blocks in the output file over which <i>dd</i> seeks, but no other portion of the output file shall be preserved. (If the size of the seek plus the size of the input file is less than the previous size of the output file, the output file shall be shortened by the copy.)                                                                                                                                                                                                                                                                                                          |
| 12770     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12771     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12772     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12773     | <b>ibs=expr</b>               | Specify the input block size, in bytes, by <i>expr</i> (default is 512).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 12774     | <b>obs=expr</b>               | Specify the output block size, in bytes, by <i>expr</i> (default is 512).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 12775     | <b>bs=expr</b>                | Set both input and output block sizes to <i>expr</i> bytes, superseding <b>ibs=</b> and <b>obs=</b> . If no conversion other than <b>sync</b> , <b>noerror</b> , and <b>notrunc</b> is specified, each input block shall be copied to the output as a single block without aggregating short blocks.                                                                                                                                                                                                                                                                                                                                                                |
| 12776     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12777     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12778     | <b>cbs=expr</b>               | Specify the conversion block size for <b>block</b> and <b>unblock</b> in bytes by <i>expr</i> (default is zero). If <b>cbs=</b> is omitted or given a value of zero, using <b>block</b> or <b>unblock</b> produces unspecified results.                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 12779     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12780     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12781 XSI |                               | The application shall ensure that this operand is also specified if the <b>conv=</b> operand is specified with a value of <b>ascii</b> , <b>ebcdic</b> , or <b>ibm</b> . For a <b>conv=</b> operand with an <b>ascii</b> value, the input is handled as described for the <b>unblock</b> value, except that characters are converted to ASCII before any trailing <space> characters are deleted. For <b>conv=</b> operands with <b>ebcdic</b> or <b>ibm</b> values, the input is handled as described for the <b>block</b> value except that the characters are converted to EBCDIC or IBM EBCDIC, respectively, after any trailing <space> characters are added.  |
| 12782     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12783     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12784     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12785     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12786     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12787     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12788     | <b>skip=n</b>                 | Skip <i>n</i> input blocks (using the specified input block size) before starting to copy. On seekable files, the implementation shall read the blocks or seek past them; on non-seekable files, the blocks shall be read and the data shall be discarded.                                                                                                                                                                                                                                                                                                                                                                                                          |
| 12789     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12790     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12791     | <b>seek=n</b>                 | Skip <i>n</i> blocks (using the specified output block size) from beginning of the output file before copying. On non-seekable files, existing blocks shall be read and space from the current end-of-file to the specified offset, if any, filled with null bytes; on seekable files, the implementation shall seek to the specified offset or read the blocks as described for non-seekable files.                                                                                                                                                                                                                                                                |
| 12792     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12793     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12794     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12795     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12796     | <b>count=n</b>                | Copy only <i>n</i> input blocks.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 12797     | <b>conv=value[,value ...]</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12798     |                               | Where <i>values</i> are comma-separated symbols from the following list:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 12799 XSI | <b>ascii</b>                  | Convert EBCDIC to ASCII; see Table 4-6 on page 340.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 12800 XSI | <b>ebcdic</b>                 | Convert ASCII to EBCDIC; see Table 4-6 on page 340.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 12801 XSI | <b>ibm</b>                    | Convert ASCII to a different EBCDIC set; see Table 4-7 on page 340.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 12802     |                               | The <b>ascii</b> , <b>ebcdic</b> , and <b>ibm</b> values are mutually-exclusive.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 12803     | <b>block</b>                  | Treat the input as a sequence of <newline> character-terminated or end-of-file-terminated variable-length records independent of the input block boundaries. Each record shall be converted to a record with a fixed length specified by the conversion block size. Any <newline> character shall be removed from the input line; <space> characters shall be appended to lines that are shorter than their conversion block size to fill the block. Lines that are longer than the conversion block size shall be truncated to the largest number of characters that fit into that size; the number of truncated lines shall be reported (see the STDERR section). |
| 12804     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12805     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12806     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12807     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12808     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12809     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12810     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12811     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 12812     |                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

|       |                                                                                                                       |                                                                                                                |
|-------|-----------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------|
| 12813 |                                                                                                                       | The <b>block</b> and <b>unblock</b> values are mutually-exclusive.                                             |
| 12814 | <b>unblock</b>                                                                                                        | Convert fixed-length records to variable length. Read a number of                                              |
| 12815 |                                                                                                                       | bytes equal to the conversion block size (or the number of bytes                                               |
| 12816 |                                                                                                                       | remaining in the input, if less than the conversion block size), delete                                        |
| 12817 |                                                                                                                       | all trailing <space> characters, and append a <newline> character.                                             |
| 12818 | <b>lcase</b>                                                                                                          | Map uppercase characters specified by the <i>LC_CTYPE</i> keyword                                              |
| 12819 |                                                                                                                       | <b>tolower</b> to the corresponding lowercase character. Characters for                                        |
| 12820 |                                                                                                                       | which no mapping is specified shall not be modified by this                                                    |
| 12821 |                                                                                                                       | conversion.                                                                                                    |
| 12822 |                                                                                                                       | The <b>lcase</b> and <b>ucase</b> symbols are mutually-exclusive.                                              |
| 12823 | <b>ucase</b>                                                                                                          | Map lowercase characters specified by the <i>LC_CTYPE</i> keyword                                              |
| 12824 |                                                                                                                       | <b>toupper</b> to the corresponding uppercase character. Characters for                                        |
| 12825 |                                                                                                                       | which no mapping is specified shall not be modified by this                                                    |
| 12826 |                                                                                                                       | conversion.                                                                                                    |
| 12827 | <b>swab</b>                                                                                                           | Swap every pair of input bytes.                                                                                |
| 12828 | <b>noerror</b>                                                                                                        | Do not stop processing on an input error. When an input error                                                  |
| 12829 |                                                                                                                       | occurs, a diagnostic message shall be written on standard error,                                               |
| 12830 |                                                                                                                       | followed by the current input and output block counts in the same                                              |
| 12831 |                                                                                                                       | format as used at completion (see the <i>STDERR</i> section). If the <b>sync</b>                               |
| 12832 |                                                                                                                       | conversion is specified, the missing input shall be replaced with null                                         |
| 12833 |                                                                                                                       | bytes and processed normally; otherwise, the input block shall be                                              |
| 12834 |                                                                                                                       | omitted from the output.                                                                                       |
| 12835 | <b>notrunc</b>                                                                                                        | Do not truncate the output file. Preserve blocks in the output file not                                        |
| 12836 |                                                                                                                       | explicitly written by this invocation of the <i>dd</i> utility. (See also the                                  |
| 12837 |                                                                                                                       | preceding <b>of=file</b> operand.)                                                                             |
| 12838 | <b>sync</b>                                                                                                           | Pad every input block to the size of the <b>ibs=</b> buffer, appending null                                    |
| 12839 |                                                                                                                       | bytes. (If either <b>block</b> or <b>unblock</b> is also specified, append <space>                             |
| 12840 |                                                                                                                       | characters, rather than null bytes.)                                                                           |
| 12841 | The behavior is unspecified if operands other than <b>conv=</b> are specified more than once.                         |                                                                                                                |
| 12842 | For the <b>bs=</b> , <b>cbs=</b> , <b>ibs=</b> , and <b>obs=</b> operands, the application shall supply an expression |                                                                                                                |
| 12843 | specifying a size in bytes. The expression, <i>expr</i> , can be:                                                     |                                                                                                                |
| 12844 | 1. A positive decimal number                                                                                          |                                                                                                                |
| 12845 | 2. A positive decimal number followed by <i>k</i> , specifying multiplication by 1 024                                |                                                                                                                |
| 12846 | 3. A positive decimal number followed by <i>b</i> , specifying multiplication by 512                                  |                                                                                                                |
| 12847 | 4. Two or more positive decimal numbers (with or without <i>k</i> or <i>b</i> ) separated by <i>x</i> , specifying    |                                                                                                                |
| 12848 | the product of the indicated values                                                                                   |                                                                                                                |
| 12849 | All of the operands are processed before any input is read.                                                           |                                                                                                                |
| 12850 | XSI                                                                                                                   | The following two tables display the octal number character values used for the <b>ascii</b> and <b>ebcdic</b> |
| 12851 |                                                                                                                       | conversions (first table) and for the <b>ibm</b> conversion (second table). In both tables, the ASCII          |
| 12852 |                                                                                                                       | values are the row and column headers and the EBCDIC values are found at their intersections.                  |
| 12853 |                                                                                                                       | For example, ASCII 0012 (LF) is the second row, third column, yielding 0045 in EBCDIC. The                     |
| 12854 |                                                                                                                       | inverted tables (for EBCDIC to ASCII conversion) are not shown, but are in one-to-one                          |
| 12855 |                                                                                                                       | correspondence with these tables. The differences between the two tables are highlighted by                    |
| 12856 |                                                                                                                       | small boxes drawn around five entries.                                                                         |

12857 **Notes to Reviewers**12858 *This section with side shading will not appear in the final copy. - Ed.*

12859 The following 2 tables are commented out of this draft to make document handling easier  
 12860 (ability to print 2-up). There are no changes to them. These diagrams are available from the  
 12861 Austin Group web site as a separate PDF file.

12862 **Table 4-6** ASCII to EBCDIC Conversion12863 **Table 4-7** ASCII to IBM EBCDIC Conversion12864 **STDIN**12865 If no **if=** operand is specified, the standard input shall be used. See the INPUT FILES section.12866 **INPUT FILES**

12867 The input file can be any file type.

12868 **ENVIRONMENT VARIABLES**12869 The following environment variables shall affect the execution of *dd*:

12870 **LANG** Provide a default value for the internationalization variables that are unset or null.  
 12871 If *LANG* is unset or null, the corresponding value from the implementation-  
 12872 dependent default locale shall be used. If any of the internationalization variables  
 12873 contains an invalid setting, the utility shall behave as if none of the variables had  
 12874 been defined.

12875 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 12876 internationalization variables.

12877 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 12878 characters (for example, single-byte as opposed to multi-byte characters in  
 12879 arguments and input files), the classification of characters as uppercase or  
 12880 lowercase, and the mapping of characters from one case to the other.

12881 **LC\_MESSAGES**

12882 Determine the locale that should be used to affect the format and contents of  
 12883 diagnostic messages written to standard error and informative messages written to  
 12884 standard output.

12885 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.12886 **ASYNCHRONOUS EVENTS**

12887 For SIGINT, the *dd* utility shall interrupt its current processing, write status information to  
 12888 standard error, and exit as though terminated by SIGINT. It shall take the standard action for all  
 12889 other signals; see the ASYNCHRONOUS EVENTS section in Section 1.11 on page 25.

12890 **STDOUT**

12891 If no **of=** operand is specified, the standard output shall be used. The nature of the output  
 12892 depends on the operands selected.

12893 **STDERR**

12894 On completion, *dd* shall write the number of input and output blocks to standard error. In the  
 12895 POSIX locale the following formats shall be used:

12896 "%u+%u records in\n", <number of whole input blocks>,  
 12897 <number of partial input blocks>

12898 "%u+%u records out\n", <number of whole output blocks>,  
 12899 <number of partial output blocks>



12900 A partial input block is one for which *read()* returned less than the input block size. A partial  
 12901 output block is one that was written with fewer bytes than specified by the output block size.

12902 In addition, when there is at least one truncated block, the number of truncated blocks shall be  
 12903 written to standard error. In the POSIX locale, the format shall be:

12904 "%u truncated %s\n", *<number of truncated blocks>*, "record" (if  
 12905 *<number of truncated blocks>* is one) "records" (otherwise)

12906 Diagnostic messages may also be written to standard error.

#### 12907 OUTPUT FILES

12908 If the **of=** operand is used, the output shall be the same as described in the STDOUT section.

#### 12909 EXTENDED DESCRIPTION

12910 None.

#### 12911 EXIT STATUS

12912 The following exit values shall be returned:

12913 0 The input file was copied successfully.

12914 >0 An error occurred.

#### 12915 CONSEQUENCES OF ERRORS

12916 If an input error is detected and the **noerror** conversion has not been specified, any partial  
 12917 output block shall be written to the output file, a diagnostic message shall be written, and the  
 12918 copy operation shall be discontinued. If some other error is detected, a diagnostic message shall  
 12919 be written and the copy operation shall be discontinued.

#### 12920 APPLICATION USAGE

12921 The input and output block size can be specified to take advantage of raw physical I/O.

12922 There are many different versions of the EBCDIC codesets. The ASCII and EBCDIC conversions  
 12923 specified for the *dd* utility perform conversions for the version specified by the tables.

#### 12924 EXAMPLES

12925 The following command:

12926 `dd if=/dev/rmt0h of=/dev/rmt1h`

12927 copies from tape drive 0 to tape drive 1, using a common historical device naming convention.

12928 The following command:

12929 `dd ibs=10 skip=1`

12930 strips the first 10 bytes from standard input.

12931 This example reads an EBCDIC tape blocked ten 80-byte EBCDIC card images per block into the  
 12932 ASCII file **x**:

12933 `dd if=/dev/tape of=x ibs=800 cbs=80 conv=ascii,lcase`

#### 12934 RATIONALE

12935 The OPTIONS section is listed as “None” because there are no options recognized by historical  
 12936 *dd* utilities. Certainly, many of the operands could have been designed to use the Utility Syntax  
 12937 Guidelines, which would have resulted in the classic hyphenated option letters. In this version  
 12938 of this volume of IEEE Std. 1003.1-200x, *dd* retains its curious JCL-like syntax due to the large  
 12939 number of applications that depend on the historical implementation.

12940 A suggested implementation technique for **conv=noerror, sync** is to zero (or <space>-fill, if  
 12941 **blocking** or **unblocking**) the input buffer before each read and to write the contents of the input

12942 buffer to the output even after an error. In this manner, any data transferred to the input buffer  
 12943 before the error was detected is preserved. Another point is that a failed read on a regular file or  
 12944 a disk generally does not increment the file offset, and *dd* must then seek past the block on which  
 12945 the error occurred; otherwise, the input error occurs repetitively. When the input is a magnetic  
 12946 tape, however, the tape normally has passed the block containing the error when the error is  
 12947 reported, and thus no seek is necessary.

12948 The default **ibs=** and **obs=** sizes are specified as 512 bytes because there are historical (largely  
 12949 portable) scripts that assume these values. If they were left unspecified, unusual results could  
 12950 occur if an implementation chose an odd block size.

12951 Historical implementations of *dd* used *creat()* when processing **of=file**. This makes the **seek=**  
 12952 operand unusable except on special files. The **conv=notrunc** feature was added because more  
 12953 recent BSD-based implementations use *open()* (without O\_TRUNC) instead of *creat()*, but they  
 12954 fail to delete output file contents after the data copied.

12955 The *w* multiplier (historically meaning *word*), is used in System V to mean 2 and in 4.2 BSD to  
 12956 mean 4. Since *word* is inherently non-portable, its use is not supported by this volume of  
 12957 IEEE Std. 1003.1-200x.

12958 Standard EBCDIC does not have the characters ' [ ' and ' ] '. The values used in the table are  
 12959 taken from a common print train that does contain them. Other than those characters, the print  
 12960 train values are not filled in, but appear to provide some of the motivation for the historical  
 12961 choice of translations reflected here.

12962 The Standard EBCDIC table provides a 1:1 translation for all 256 bytes.

12963 The IBM EBCDIC table does not provide such a translation. The marked cells in the tables differ  
 12964 in such a way that:

- 12965 1. EBCDIC 0112 ( ' ¢ ' ) and 0152 (broken pipe) do not appear in the table.
- 12966 2. EBCDIC 0137 ( ' ¬ ' ) translates to/from ASCII 0236 ( ' ^ ' ). In the standard table, EBCDIC  
 12967 0232 (no graphic) is used.
- 12968 3. EBCDIC 0241 ( ' ~ ' ) translates to/from ASCII 0176 ( ' ~ ' ). In the standard table, EBCDIC  
 12969 0137 ( ' ¬ ' ) is used.
- 12970 4. 0255 ( ' [ ' ) and 0275 ( ' ] ' ) appear twice, once in the same place as for the standard table  
 12971 and once in place of 0112 ( ' ¢ ' ) and 0241 ( ' ~ ' ).

12972 In net result:

12973 EBCDIC 0275 ( ' ] ' ) displaced EBCDIC 0241 ( ' ~ ' ) in cell 0345.

12974 That displaced EBCDIC 0137 ( ' ¬ ' ) in cell 0176.

12975 That displaced EBCDIC 0232 (no graphic) in cell 0136.

12976 That replaced EBCDIC 0152 (broken pipe) in cell 0313.

12977 EBCDIC 0255 ( ' [ ' ) replaced EBCDIC 0112 ( ' ¢ ' ).

12978 This translation, however, reflects historical practice that (ASCII) ' ~ ' and ' ¬ ' were often  
 12979 mapped to each other, as were ' [ ' and ' ¢ ' ; and ' ] ' and (EBCDIC) ' ~ ' .

12980 The **chs** operand is required if any of the **ascii**, **ebcdic**, or **ibm** operands are specified. For the  
 12981 **ascii** operand, the input is handled as described for the **unblock** operand except that characters  
 12982 are converted to ASCII before the trailing <space>s are deleted. For the **ebcdic** and **ibm**  
 12983 operands, the input is handled as described for the **block** operand except that the characters are  
 12984 converted to EBCDIC or IBM EBCDIC after the trailing <space>s are added.

- 12985 The **block** and **unblock** keywords are from historical BSD practice.
- 12986 The consistent use of the word **record** in standard error messages matches most historical  
12987 practice. An earlier version of System V used **block**, but this has been updated in more recent  
12988 releases.
- 12989 Early proposals only allowed two numbers separated by **x** to be used in a product when  
12990 specifying **bs=**, **cbs=**, **ibs=**, and **obs=** sizes. This was changed to reflect the historical practice of  
12991 allowing multiple numbers in the product as provided by Version 7 and all releases of System V  
12992 and BSD.
- 12993 A change to the *swab* conversion is required to match historical practice and is the result of PASC  
12994 Interpretation 1003.2-92 #03 and #04, submitted for the ISO POSIX-2: 1993 standard.
- 12995 A change to the handling of SIGINT is required to match historical practice and is the result of  
12996 PASC Interpretation 1003.2-92 #06 submitted for the ISO POSIX-2: 1993 standard.
- 12997 **FUTURE DIRECTIONS**
- 12998 None.
- 12999 **SEE ALSO**
- 13000 *sed*, *tr*
- 13001 **CHANGE HISTORY**
- 13002 First released in Issue 2.
- 13003 **Issue 4**
- 13004 Aligned with the ISO/IEC 9945-2: 1993 standard.
- 13005 **Issue 5**
- 13006 The second paragraph of the **cbs=** description is reworded and marked EX.
- 13007 FUTURE DIRECTIONS section added.
- 13008 **Issue 6**
- 13009 Changes are made to *swab* conversion and SIGINT handling to align with the IEEE P1003.2b  
13010 draft standard.
- 13011 The normative text is reworded to avoid use of the term “must” for application requirements.

## 13012 NAME

13013       delta — make a delta (change) to an SCCS file (**DEVELOPMENT**)

## 13014 SYNOPSIS

13015 xSI       delta [-nps][-g *list*][-m *mrlist*][-r *SID*][-y[*comment*]] *file...*

13016

## 13017 DESCRIPTION

13018       The *delta* utility shall be used to permanently introduce into the named SCCS files changes that  
13019       were made to the files retrieved by *get* (called the *g-files*, or generated files).

## 13020 OPTIONS

13021       The *delta* utility shall conform to the System Interface Definitions volume of  
13022       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines, except that the *-y* option has an  
13023       optional option-argument. This optional option-argument cannot be presented as a separate  
13024       argument.

13025       The following options shall be supported:

13026       *-r SID*       Uniquely identify which delta is to be made to the SCCS file. The use of this option  
13027       is necessary only if two or more outstanding *get* commands for editing (*get -e*) on  
13028       the same SCCS file were done by the same person (login name). The *SID* value  
13029       specified with the *-r* option can be either the *SID* specified on the *get* command  
13030       line or the *SID* to be made as reported by the *get* utility; see *get* on page 510.

13031       *-s*           Suppress the report to standard output of the activity associated with each *file*.  
13032       See the STDOUT section.

13033       *-n*           Specify retention of the edited *g-file* (normally removed at completion of delta  
13034       processing).

13035       *-g list*       Specify a *list*, (see *get* on page 510 for the definition of *list*) of deltas that shall be  
13036       ignored when the file is accessed at the change level (*SID*) created by this delta.

13037       *-m mrlist*   Specify a modification request (MR) number that the application shall supply as  
13038       the reason for creating the new delta. This is used if the SCCS file has the *-v* flag  
13039       set; see *admin* on page 160.

13040       If *-m* is not used and the standard input is a terminal, the prompt described in the  
13041       STDOUT section is written to standard output before the standard input is read; if  
13042       the standard input is not a terminal, no prompt is issued.

13043       MRs in a list are separated by <blank>s. An unescaped <newline> character  
13044       terminates the MR list.

13045       Note that if the *-v* flag has a value, it is taken to be the name of a program which  
13046       validates the correctness of the MR numbers. If a non-zero exit status is returned  
13047       from the MR number validation program, *delta* terminates. (It is assumed that the  
13048       MR numbers were not all valid.)

13049       *-y[comment]* Describe the reason for making the delta. This is an arbitrary group of lines that  
13050       would meet the definition of a text file. Systems support *comments* from zero to 512  
13051       bytes and may support longer values. A null string (specified as either *-y*, *-y* " ", or  
13052       in response to a prompt for a comment) is considered a valid *comment*.

13053       If *-y* is not specified and the standard input is a terminal, the prompt described in  
13054       the STDOUT section is written to standard output before the standard input is  
13055       read; if the standard input is not a terminal, no prompt is issued. An unescaped  
13056       <newline> character terminates the comment text.

13057                   The **-y** option is required if the *file* operand is specified as **'-'**.

13058           **-p**           Write (to standard output) the SCCS file differences before and after the delta is  
13059                   applied in *diff* format; see *diff* on page 352.

## 13060 OPERANDS

13061           The following operand shall be supported:

13062           *file*           A path name of an existing SCCS file or a directory. If *file* is a directory, *delta*  
13063                   behaves as though each file in the directory were specified as a named file, except  
13064                   that non-SCCS files (last component of the path name does not begin with **s.**) and  
13065                   unreadable files are silently ignored.

13066                   If a single instance *file* is specified as **'-'**, the standard input is read; each line of  
13067                   the standard input is taken to be the name of an SCCS file to be processed. Non-  
13068                   SCCS files and unreadable files are silently ignored.

## 13069 STDIN

13070           The standard input is a text file used only in the following cases:

- 13071           • To read an *mrlist* or a *command* (see the **-m** and **-y** options).
- 13072           • A *file* operand is specified as **'-'**.

## 13073 INPUT FILES

13074           Input files are text files whose data is to be included in the SCCS files. If the first character of any  
13075           line of an input file is SOH (binary 001), the results are unspecified.

## 13076 ENVIRONMENT VARIABLES

13077           The following environment variables shall affect the execution of *delta*:

13078           **LANG**           Provide a default value for the internationalization variables that are unset or null.  
13079                   If **LANG** is unset or null, the corresponding value from the implementation-  
13080                   dependent default locale shall be used. If any of the internationalization variables  
13081                   contains an invalid setting, the utility shall behave as if none of the variables had  
13082                   been defined.

13083           **LC\_ALL**          If set to a non-empty string value, override the values of all the other  
13084                   internationalization variables.

13085           **LC\_CTYPE**       Determine the locale for the interpretation of sequences of bytes of text data as  
13086                   characters (for example, single-byte as opposed to multi-byte characters in  
13087                   arguments and input files).

13088           **LC\_MESSAGES**   Determine the locale that should be used to affect the format and contents of  
13089                   diagnostic messages written to standard error, and informative messages written  
13090                   to standard output.  
13091

13092           **NLSPATH**       Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

## 13093 ASYNCHRONOUS EVENTS

13094           Default.

## 13095 STDOUT

13096           The standard output shall be used only for the following messages in the POSIX locale:

- 13097           • Prompts (see the **-m** and **-y** options) in the following formats:

13098           "MRs? "

13099 "comments? "

13100 The MR prompt, if written, always precedes the comments prompt.

13101 • A report of each *file*'s activities (unless the `-s` option is specified) in the following format:

13102 "%s\n%d inserted\n%d deleted\n%d unchanged\n", <New SID>,  
13103 <number of lines inserted>, <number of lines deleted>,  
13104 <number of lines unchanged>

#### 13105 **STDERR**

13106 Used only for diagnostic messages.

#### 13107 **OUTPUT FILES**

13108 Any SCCS files updated are files of an unspecified format.

#### 13109 **EXTENDED DESCRIPTION**

13110 None.

#### 13111 **EXIT STATUS**

13112 The following exit values shall be returned:

13113 0 Successful completion.

13114 >0 An error occurred.

#### 13115 **CONSEQUENCES OF ERRORS**

13116 Default.

#### 13117 **APPLICATION USAGE**

13118 Application writers should note that this utility need not be provided on systems that do not  
13119 support the XSI Development Utilities option.

#### 13120 **EXAMPLES**

13121 None.

#### 13122 **RATIONALE**

13123 None.

#### 13124 **FUTURE DIRECTIONS**

13125 A version of *delta* that fully supports the System Interface Definitions volume of  
13126 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines may be introduced in a future  
13127 version.

#### 13128 **SEE ALSO**

13129 *admin, diff, get, prs, rmdel*

#### 13130 **CHANGE HISTORY**

13131 First released in Issue 2.

#### 13132 **Issue 4**

13133 Format reorganized.

13134 Exceptions to Utility Syntax Guidelines conformance noted.

13135 Internationalized environment variable support mandated.

#### 13136 **Issue 5**

13137 The output format description in the STDOUT section is corrected.

13138 **Issue 6**

13139 The APPLICATION USAGE section is added.

13140 The normative text is reworded to avoid use of the term “must” for application requirements.

13141 **NAME**

13142       df — report free disk space

13143 **SYNOPSIS**13144 UP XSI   df [-k][-P|-t][*file...*]

13145

13146 **DESCRIPTION**

13147 XSI       The *df* utility shall write the amount of available space and file slots for file systems on which the  
 13148       invoking user has appropriate read access. File systems shall be specified by the *file* operands;  
 13149       when none are specified, information shall be written for all file systems. The format of the  
 13150       default output from *df* is unspecified, but all space figures are reported in 512-byte units, unless  
 13151       the **-k** option is specified. This output shall contain at least the file system names, amount of  
 13152 XSI       available space on each of these file systems, and the number of free file slots, or *inodes*,  
 13153       available; when **-t** is specified, the output contains the total allocated space as well.

13154 **OPTIONS**

13155       The *df* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 13156       Section 12.2, Utility Syntax Guidelines.

13157       The following options shall be supported:

13158       **-k**           Use 1024-byte units, instead of the default 512-byte units, when writing space  
 13159                      figures.

13160       **-P**           Produce output in the format described in the STDOUT section.

13161 XSI       **-t**           Include total allocated-space figures in the output.

13162 **OPERANDS**

13163       The following operand shall be supported:

13164       *file*        A path name of a file within the hierarchy of the desired file system. If a file other  
 13165 XSI       than a FIFO, a regular file, a directory or a special file representing the device  
 13166       containing the file system (for example, **/dev/dsk/0s1**) is specified, the results are  
 13167       unspecified. Otherwise, *df* shall write the amount of free space in the file system  
 13168       containing the specified *file* operand.

13169 **STDIN**

13170       Not used.

13171 **INPUT FILES**

13172       None.

13173 **ENVIRONMENT VARIABLES**13174       The following environment variables shall affect the execution of *df*:

13175       **LANG**        Provide a default value for the internationalization variables that are unset or null.  
 13176       If **LANG** is unset or null, the corresponding value from the implementation-  
 13177       dependent default locale shall be used. If any of the internationalization variables  
 13178       contains an invalid setting, the utility shall behave as if none of the variables had  
 13179       been defined.

13180       **LC\_ALL**       If set to a non-empty string value, override the values of all the other  
 13181       internationalization variables.

13182       **LC\_CTYPE**   Determine the locale for the interpretation of sequences of bytes of text data as  
 13183       characters (for example, single-byte as opposed to multi-byte characters in  
 13184       arguments).



13185 **LC\_MESSAGES**  
 13186 Determine the locale that should be used to affect the format and contents of  
 13187 diagnostic messages written to standard error and informative messages written to  
 13188 standard output.

13189 XSI **NLSPATH** Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

13190 **ASYNCHRONOUS EVENTS**  
 13191 Default.

13192 **STDOUT**  
 13193 When both the **-k** and **-P** options are specified, the following header line shall be written (in the  
 13194 POSIX locale):  
 13195 "Filesystem 1024-blocks Used Available Capacity Mounted on\n"  
 13196 When the **-P** option is specified without the **-k** option, the following header line shall be written  
 13197 (in the POSIX locale):  
 13198 "Filesystem 512-blocks Used Available Capacity Mounted on\n"  
 13199 The implementation may adjust the spacing of the header line and the individual data lines so  
 13200 that the information is presented in orderly columns.  
 13201 The remaining output with **-P** shall consist of one line of information for each specified file  
 13202 system. These lines shall be formatted as follows:  
 13203 "%s %d %d %d %d%% %s\n", <file system name>, <total space>,  
 13204 <space used>, <space free>, <percentage used>,  
 13205 <file system root>  
 13206 In the following list, all quantities expressed in 512-byte units (1 024-byte when **-k** is specified)  
 13207 shall be rounded up to the next higher unit. The fields are:  
 13208 <file system name>  
 13209 The name of the file system, in an implementation-dependent format.  
 13210 <total space> The total size of the file system in 512-byte units. The exact meaning of this figure  
 13211 is implementation-dependent, but should include <space used>, <space free>, plus  
 13212 any space reserved by the system not normally available to a user.  
 13213 <space used> The total amount of space allocated to existing files in the file system, in 512-byte  
 13214 units.  
 13215 <space free> The total amount of space available within the file system for the creation of new  
 13216 files by unprivileged users, in 512-byte units. When this figure is less than or equal  
 13217 to zero, it shall not be possible to create any new files on the file system without  
 13218 first deleting others, unless the process has appropriate privileges. The figure  
 13219 written may be less than zero.  
 13220 <percentage used>  
 13221 The percentage of the normally available space that is currently allocated to all  
 13222 files on the file system. This shall be calculated using the fraction:  
 13223 <space used>/(<space used>+ <space free>)  
 13224 expressed as a percentage. This percentage may be greater than 100 if <space free>  
 13225 is less than zero. The percentage value shall be expressed as a positive integer,  
 13226 with any fractional result causing it to be rounded to the next highest integer.

13227           <file system root>  
 13228           The directory below which the file system hierarchy appears.

13229 XSI       The output format is unspecified when **-t** is used.

#### 13230 **STDERR**

13231       Used only for diagnostic messages.

#### 13232 **OUTPUT FILES**

13233       None.

#### 13234 **EXTENDED DESCRIPTION**

13235       None.

#### 13236 **EXIT STATUS**

13237       The following exit values shall be returned:

13238       0   Successful completion.

13239       >0   An error occurred.

#### 13240 **CONSEQUENCES OF ERRORS**

13241       Default.

#### 13242 **APPLICATION USAGE**

13243       On most systems, the “name of the file system, in an implementation-dependent format” is the  
 13244       special file on which the file system is mounted.

13245       On large file systems, the calculation specified for percentage used can create huge rounding  
 13246       errors.

13247       Application writers should note that this utility need not be provided on systems that do not  
 13248       support the User Portability Utilities option.

#### 13249 **EXAMPLES**

13250       1. The following example writes portable information about the **/usr** file system:

13251           df -P /usr

13252       2. Assuming that **/usr/src** is part of the **/usr** file system, the following produces the same  
 13253       output as the previous example:

13254           df -P /usr/src

#### 13255 **RATIONALE**

13256       The behavior of *df* with the **-P** option is the default action of the 4.2 BSD *df* utility. The uppercase  
 13257       **-P** was selected to avoid collision with a known industry extension using **-p**.

13258       Historical *df* implementations vary considerably in their default output. It was therefore  
 13259       necessary to describe the default output in a loose manner to accommodate all known historical  
 13260       implementations and to add a portable option (**-P**) to provide information in a portable format.

13261       The use of 512-byte units is historical practice and maintains compatibility with *ls* and other  
 13262       utilities in this volume of IEEE Std. 1003.1-200x. This does not mandate that the file system itself  
 13263       be based on 512-byte blocks. The **-k** option was added as a compromise measure. It was agreed  
 13264       by the standard developers that 512 bytes was the best default unit because of its complete  
 13265       historical consistency on System V (*versus* the mixed 512/1 024-byte usage on BSD systems), and  
 13266       that a **-k** option to switch to 1 024-byte units was a good compromise. Users who prefer the  
 13267       more logical 1 024-byte quantity can easily alias *df* to *df -k* without breaking many historical  
 13268       scripts relying on the 512-byte units.

13269 It was suggested that *df* and the various related utilities be modified to access a *BLOCKSIZE*  
13270 environment variable to achieve consistency and user acceptance. Since this is not historical  
13271 practice on any system, it is left as a possible area for system extensions and will be re-evaluated  
13272 in a future version if it is widely implemented.

13273 **FUTURE DIRECTIONS**

13274 None.

13275 **SEE ALSO**

13276 *find*

13277 **CHANGE HISTORY**

13278 First released in Issue 2.

13279 **Issue 4**

13280 Aligned with the ISO/IEC 9945-2:1993 standard.

13281 **Issue 6**

13282 This utility is now marked as part of the User Portability Utilities option.

## 13283 NAME

13284 diff — compare two files

## 13285 SYNOPSIS

13286 MAN diff [-c | -e | -f | -C n][-br] file1 file2

## 13287 DESCRIPTION

13288 The *diff* utility shall compare the contents of *file1* and *file2* and write to standard output a list of  
 13289 changes necessary to convert *file1* into *file2*. This list should be minimal. No output shall be  
 13290 produced if the files are identical.

## 13291 OPTIONS

13292 The *diff* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 13293 Section 12.2, Utility Syntax Guidelines.

13294 The following options shall be supported:

13295 **-b** Cause any amount of white space at the end of a line to be treated as a single  
 13296 <newline> character (that is, the white-space characters preceding the <newline>  
 13297 character are ignored) and other strings of white-space characters, not including  
 13298 <newline> characters, to compare equal.

13299 **-c** Produce output in a form that provides three lines of context.

13300 **-C n** Produce output in a form that provides *n* lines of context (where *n* shall be  
 13301 interpreted as a positive decimal integer).

13302 **-e** Produce output in a form suitable as input for the *ed* utility, which can then be  
 13303 used to convert *file1* into *file2*.

13304 MAN **-f** Produce output in an alternative form, similar in format to **-e**, but unsuitable as  
 13305 input for the *ed* utility, and in the opposite order.

13306 **-r** Apply *diff* recursively to files and directories of the same name when *file1* and *file2*  
 13307 are both directories.

## 13308 OPERANDS

13309 The following operands shall be supported:

13310 *file1, file2* A path name of a file to be compared. If either the *file1* or *file2* operand is '-', the  
 13311 standard input shall be used in its place.

13312 If both *file1* and *file2* are directories, *diff* shall not compare block special files, character special  
 13313 files, or FIFO special files to any files and shall not compare regular files to directories. The  
 13314 system documentation shall specify the behavior of *diff* on implementation-dependent file types  
 13315 not specified by the System Interfaces volume of IEEE Std. 1003.1-200x when found in  
 13316 directories. Further details are as specified in **Diff Directory Comparison Format** on page 353.

13317 If only one of *file1* and *file2* is a directory, *diff* shall be applied to the non-directory file and the file  
 13318 contained in the directory file with a file name that is the same as the last component of the non-  
 13319 directory file.

## 13320 STDIN

13321 The standard input shall be used only if one of the *file1* or *file2* operands references standard  
 13322 input. See the INPUT FILES section.

## 13323 INPUT FILES

13324 The input files shall be text files.

13325 **ENVIRONMENT VARIABLES**

13326 The following environment variables shall affect the execution of *diff*:

13327 **LANG** Provide a default value for the internationalization variables that are unset or null.  
 13328 If *LANG* is unset or null, the corresponding value from the implementation-  
 13329 dependent default locale shall be used. If any of the internationalization variables  
 13330 contains an invalid setting, the utility shall behave as if none of the variables had  
 13331 been defined.

13332 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 13333 internationalization variables.

13334 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 13335 characters (for example, single-byte as opposed to multi-byte characters in  
 13336 arguments and input files).

13337 **LC\_MESSAGES**  
 13338 Determine the locale that should be used to affect the format and contents of  
 13339 diagnostic messages written to standard error and informative messages written to  
 13340 standard output.

13341 **LC\_TIME** Determine the locale for affecting the format of file timestamps written with the  
 13342 **-C** and **-c** options.

13343 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

13344 **TZ** Determine the locale for affecting the timezone used for calculating file  
 13345 timestamps written with the **-C** and **-c** options.

13346 **ASYNCHRONOUS EVENTS**

13347 Default.

13348 **STDOUT**13349 **Diff Directory Comparison Format**

13350 If both *file1* and *file2* are directories, the following output formats shall be used.

13351 In the POSIX locale, each file that is present in only one directory shall be reported using the  
 13352 following format:

13353 "Only in %s: %s\n", <directory pathname>, <filename>

13354 In the POSIX locale, subdirectories that are common to the two directories may be reported with  
 13355 the following format:

13356 "Common subdirectories: %s and %s\n", <directory1 pathname>,  
 13357 <directory2 pathname>

13358 For each file common to the two directories if the two files are not to be compared, the following  
 13359 format shall be used in the POSIX locale:

13360 "File %s is a %s while file %s is a %s\n", <directory1 pathname>,  
 13361 <file type of directory1 pathname>, <directory2 pathname>,  
 13362 <file type of directory2 pathname>

13363 For each file common to the two directories, if the files are compared and are identical, no output |  
 13364 shall be written. If the two files differ, the following format is written:

13365 "diff %s %s %s\n", <diff\_options>, <filename1>, <filename2>

13366 where *<diff\_options>* are the options as specified on the command line. Depending on these  
 13367 options, one of the following output formats shall be used to write the differences.

13368 All directory path names listed in this section shall be relative to the original command line  
 13369 arguments. All other names of files listed in this section are file names (path name components).

### 13370 Diff Default Output Format

13371 MAN The default (without **-e**, **-f**, **-c**, or **-C** options) *diff* utility output shall contain lines of these forms:

13372 "%da%d\n", <num1>, <num2>

13373 "%da%d,%d\n", <num1>, <num2>, <num3>

13374 "%dd%d\n", <num1>, <num2>

13375 "%d,%dd%d\n", <num1>, <num2>, <num3>

13376 "%dc%d\n", <num1>, <num2>

13377 "%d,%dc%d\n", <num1>, <num2>, <num3>

13378 "%dc%d,%d\n", <num1>, <num2>, <num3>

13379 "%d,%dc%d,%d\n", <num1>, <num2>, <num3>, <num4>

13380 These lines resemble *ed* subcommands to convert *file1* into *file2*. The line numbers before the  
 13381 action letters shall pertain to *file1*; those after shall pertain to *file2*. Thus, by exchanging *a* for *d*  
 13382 and reading the line in reverse order, one can also determine how to convert *file2* into *file1*. As in  
 13383 *ed*, identical pairs (where *num1*= *num2*) are abbreviated as a single number.

13384 Following each of these lines, *diff* shall write to standard output all lines affected in the first file  
 13385 using the format:

13386 "<Δ%s", <line>

13387 and all lines affected in the second file using the format:

13388 ">Δ%s", <line>

13389 If there are lines affected in both *file1* and *file2* (as with the *c* subcommand), the changes are  
 13390 separated with a line consisting of three hyphens:

13391 "——\n"

### 13392 Diff -e Output Format

13393 With the **-e** option, a script shall be produced that shall, when provided as input to *ed*, along  
 13394 with an appended **w** (write) command, convert *file1* into *file2*. Only the **a** (append), **c** (change), **d**  
 13395 (delete), **i** (insert), and **s** (substitute) commands of *ed* shall be used in this script. Text lines,  
 13396 except those consisting of the single character period ( '.' ), shall be output as they appear in the  
 13397 file.

13398 **Diff -f Output Format**

13399 MAN With the **-f** option, an alternative format of script shall be produced. It is similar to that  
 13400 produced by **-e**, with the following differences:

- 13401 1. It is expressed in reverse sequence; the output of **-e** orders changes from the end of the file  
 13402 to the beginning; the **-f** from beginning to end.
- 13403 2. The command form `<lines> <command-letter>` used by **-e** is reversed. For example,  
 13404 `10c` with **-e** would be `c10` with **-f**.
- 13405 3. The form used for ranges of line numbers is `<space>` character-separated, rather than  
 13406 comma-separated.

13407 **Diff -c or -C Output Format**

13408 With the **-c** or **-C** option, the output format shall consist of affected lines along with  
 13409 surrounding lines of context. The affected lines shall show which ones need to be deleted or  
 13410 changed in *file1*, and those added from *file2*. With the **-c** option, three lines of context, if  
 13411 available, shall be written before and after the affected lines. With the **-C** option, the user can  
 13412 specify how many lines of context are written. The exact format follows.

13413 The name and last modification time of each file shall be output in the following format:

```
13414 "*** %s %s\n", file1, <file1 time stamp>
13415 "—— %s %s\n", file2, <file2 time stamp>
```

13416 Each `<file>` field shall be the path name of the corresponding file being compared. The path  
 13417 name written for standard input is unspecified.

13418 In the POSIX locale, each `<timestamp>` field shall be equivalent to the output from the following  
 13419 command:

```
13420 date "+%a %b %e %T %Y"
```

13421 without the trailing `<newline>` character, executed at the time of last modification of the  
 13422 corresponding file (or the current time, if the file is standard input).

13423 Then, the following output formats shall be applied for every set of changes.

13424 First, a line shall be written in the following format:

```
13425 "*****\n"
```

13426 Next, the range of lines in *file1* shall be written in the following format:

```
13427 "*** %d,%d ***\n", <beginning line number>, <ending line number>
```

13428 Next, the affected lines along with lines of context (unaffected lines) shall be written. Unaffected  
 13429 lines shall be written in the following format:

```
13430 "ΔΔ%s", <unaffected_line>
```

13431 Deleted lines shall be written as:

```
13432 "-Δ%s", <deleted_line>
```

13433 Changed lines shall be written as:

```
13434 "!Δ%s", <changed_line>
```

13435 Next, the range of lines in *file2* shall be written in the following format:

13436 "—— %d,%d ——\n", <beginning line number>, <ending line number>

13437 Then, lines of context and changed lines shall be written as described in the previous formats.

13438 Lines added from *file2* shall be written in the following format:

13439 "+Δ%s", <added\_line>

13440 **STDERR**

13441 Used only for diagnostic messages.

13442 **OUTPUT FILES**

13443 None.

13444 **EXTENDED DESCRIPTION**

13445 None.

13446 **EXIT STATUS**

13447 The following exit values shall be returned:

13448 0 No differences were found.

13449 1 Differences were found.

13450 >1 An error occurred.

13451 **CONSEQUENCES OF ERRORS**

13452 Default.

13453 **APPLICATION USAGE**

13454 If lines at the end of a file are changed and other lines are added, *diff* output may show this as a delete and add, as a change, or as a change and add; *diff* is not expected to know which happened and users should not care about the difference in output as long as it clearly shows the differences between the files.

13458 **EXAMPLES**

13459 If **dir1** is a directory containing a directory named **x**, **dir2** is a directory containing a directory

13460 named **x**, **dir1/x** and **dir2/x** both contain files named **date.out**, and **dir2/x** contains a file named **y**,

13461 the command:

13462 `diff -r dir1 dir2`

13463 could produce output similar to:

13464 Common subdirectories: dir1/x and dir2/x

13465 Only in dir2/x: y

13466 `diff -r dir1/x/date.out dir2/x/date.out`

13467 `lcl`

13468 `< Mon Jul 2 13:12:16 PDT 1990`

13469 `——`

13470 `> Tue Jun 19 21:41:39 PDT 1990`

13471 **RATIONALE**

13472 The **-h** option was omitted because it was insufficiently specified and does not add to application portability.

13474 Historical implementations employ algorithms that do not always produce a minimum list of

13475 differences; the current language about making every effort is the best this volume of

13476 IEEE Std. 1003.1-200x can do, as there is no metric that could be employed to judge the quality of

13477 implementations against any and all file contents. The statement "This list should be minimal"

13478 clearly implies that implementations are not expected to provide the following output when

13479 comparing two 100-line files that differ in only one character on a single line:



```

13480 1,100c1,100
13481 all 100 lines from file1 preceded with "< "
13482 ———
13483 all 100 lines from file2 preceded with "> "

```

13484 The “Only in” messages required when the `-r` option is specified are not used by most historical  
 13485 implementations if the `-e` option is also specified. It is required here because it provides useful  
 13486 information that must be provided to update a target directory hierarchy to match a source  
 13487 hierarchy. The “Common subdirectories” messages are written by System V and 4.3 BSD when  
 13488 the `-r` option is specified. They are allowed here but are not required because they are reporting  
 13489 on something that is the same, not reporting a difference, and are not needed to update a target  
 13490 hierarchy.

13491 The `-c` option, which writes output in a format using lines of context, has been included. The  
 13492 format is useful for a variety of reasons, among them being much improved readability and the  
 13493 ability to understand difference changes when the target file has line numbers that differ from  
 13494 another similar, but slightly different, copy. The *patch* utility is most valuable when working  
 13495 with difference listings using the context format. The BSD version of `-c` takes an optional  
 13496 argument specifying the amount of context. Rather than overloading `-c` and breaking the Utility  
 13497 Syntax Guidelines for *diff*, the standard developers decided to add a separate option for  
 13498 specifying a context diff with a specified amount of context (`-C`). Also, the format for context  
 13499 *diffs* was extended slightly in 4.3 BSD to allow multiple changes that are within context lines  
 13500 from each other to be merged together. The output format contains an additional four asterisks  
 13501 after the range of affected lines in the first file name. This was to provide a flag for old programs  
 13502 (like old versions of *patch*) that only understand the old context format. The version of context  
 13503 described here does not require that multiple changes within context lines be merged, but it does  
 13504 not prohibit it either. The extension is upward-compatible, so any vendors that wish to retain the  
 13505 old version of *diff* can do so by adding the extra four asterisks (that is, utilities that currently use  
 13506 *diff* and understand the new merged format will also understand the old unmerged format, but  
 13507 not *vice versa*).

13508 The substitute command was added as an additional format for the `-e` option. This was added to  
 13509 provide implementations a way to fix the classic “dot alone on a line” bug present in many  
 13510 versions of *diff*. Since many implementations have fixed this bug, the standard developers  
 13511 decided not to standardize broken behavior, but rather to provide the necessary tool for fixing  
 13512 the bug. One way to fix this bug is to output two periods whenever a lone period is needed, then  
 13513 terminate the append command with a period, and then use the substitute command to convert  
 13514 the two periods into one period.

13515 The BSD-derived `-r` option was added to provide a mechanism for using *diff* to compare two file  
 13516 system trees. This behavior is useful, is standard practice on all BSD-derived systems, and is not  
 13517 easily reproducible with the *find* utility.

13518 The requirement that *diff* not compare files in some circumstances, even though they have the  
 13519 same name, is based on the actual output of historical implementations. The message specified  
 13520 here is already in use when a directory is being compared to a non-directory. It is extended here  
 13521 to preclude the problems arising from running into FIFOs and other files that would cause *diff* to  
 13522 hang waiting for input with no indication to the user that *diff* was hung. In most common usage,  
 13523 *diff -r* should indicate differences in the file hierarchies, not the difference of contents of devices  
 13524 pointed to by the hierarchies.

13525 Many early implementations of *diff* require seekable files. Since the System Interfaces volume of  
 13526 IEEE Std. 1003.1-200x supports named pipes, the standard developers decided that such a  
 13527 restriction was unreasonable. Note also that the allowed file name – almost always refers to a  
 13528 pipe.

13529 No directory search order is specified for *diff*. The historical ordering is, in fact, not optimal, in  
 13530 that it prints out all of the differences at the current level, including the statements about all  
 13531 common subdirectories before recursing into those subdirectories.

13532 The message:

13533 "diff %s %s %s\n", <diff\_options>, <filename1>, <filename2>

13534 does not vary by locale because it is the representation of a command, not an English sentence.

13535 **FUTURE DIRECTIONS**

13536 None.

13537 **SEE ALSO**

13538 *cmp, comm, ed*

13539 **CHANGE HISTORY**

13540 First released in Issue 2.

13541 **Issue 4**

13542 Aligned with the ISO/IEC 9945-2:1993 standard.

13543 **Issue 5**

13544 FUTURE DIRECTIONS section added.

13545 **Issue 6**

13546 The following new requirements on POSIX implementations derive from alignment with the  
 13547 Single UNIX Specification:

13548 • The **-f** option is added.

13549 The output format for **-c** or **-C** format is changed to align with changes to the IEEE P1003.2b  
 13550 draft standard resulting from PASC Interpretation 1003.2-92 #71.

13551 The normative text is reworded to avoid use of the term “must” for application requirements.

13552 **NAME**

13553           dirname — return the directory portion of path name

13554 **SYNOPSIS**13555           dirname *string*13556 **DESCRIPTION**

13557       The *string* operand shall be treated as a path name, as defined in the System Interface Definitions  
 13558       volume of IEEE Std. 1003.1-200x, Section 3.272, Path Name. The string *string* shall be converted  
 13559       to the name of the directory containing the file name corresponding to the last path name  
 13560       component in *string*, performing actions equivalent to the following steps in order:

- 13561           1. If *string* is //, skip steps 2 to 5.
- 13562           2. If *string* consists entirely of slash characters, *string* shall be set to a single slash character. In  
 13563           this case, skip steps 3 to 8.
- 13564           3. If there are any trailing slash characters in *string*, they shall be removed.
- 13565           4. If there are no slash characters remaining in *string*, *string* shall be set to a single period  
 13566           character. In this case, skip steps 5 to 8.
- 13567           5. If there are any trailing non-slash characters in *string*, they shall be removed.
- 13568           6. If the remaining *string* is //, it is implementation-dependent whether steps 7 and 8 are  
 13569           skipped or processed.
- 13570           7. If there are any trailing slash characters in *string*, they shall be removed.
- 13571           8. If the remaining *string* is empty, *string* shall be set to a single slash character.

13572       The resulting string shall be written to standard output.

13573 **OPTIONS**

13574       None.

13575 **OPERANDS**

13576       The following operand shall be supported:

13577       *string*           A string.13578 **STDIN**

13579       Not used.

13580 **INPUT FILES**

13581       None.

13582 **ENVIRONMENT VARIABLES**13583       The following environment variables shall affect the execution of *dirname*:

- |                                           |                 |                                                                                                                                                                                                                                                                                                                                                                 |
|-------------------------------------------|-----------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 13584<br>13585<br>13586<br>13587<br>13588 | <i>LANG</i>     | Provide a default value for the internationalization variables that are unset or null. If <i>LANG</i> is unset or null, the corresponding value from the implementation-dependent default locale will be used. If any of the internationalization variables contains an invalid setting, the utility shall behave as if none of the variables had been defined. |
| 13589<br>13590                            | <i>LC_ALL</i>   | If set to a non-empty string value, override the values of all the other internationalization variables.                                                                                                                                                                                                                                                        |
| 13591<br>13592<br>13593                   | <i>LC_CTYPE</i> | Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).                                                                                                                                                                                       |

13594 **LC\_MESSAGES**

13595 Determine the locale that should be used to affect the format and contents of  
 13596 diagnostic messages written to standard error.

13597 XSI **NLSPATH** Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

13598 **ASYNCHRONOUS EVENTS**

13599 Default.

13600 **STDOUT**

13601 The *dirname* utility shall write a line to the standard output in the following format:

13602 "%s\n", <resulting string>

13603 **STDERR**

13604 Used only for diagnostic messages.

13605 **OUTPUT FILES**

13606 None.

13607 **EXTENDED DESCRIPTION**

13608 None.

13609 **EXIT STATUS**

13610 The following exit values shall be returned:

13611 0 Successful completion.

13612 >0 An error occurred.

13613 **CONSEQUENCES OF ERRORS**

13614 Default.

13615 **APPLICATION USAGE**

13616 The definition of *pathname* specifies implementation-dependent behavior for path names starting  
 13617 with two slash characters. Therefore, applications shall not arbitrarily add slashes to the  
 13618 beginning of a path name unless they can ensure that there are more or less than two or are  
 13619 prepared to deal with the implementation-dependent consequences.

13620 **EXAMPLES**

| Command                 | Results     |
|-------------------------|-------------|
| <i>dirname</i> /        | /           |
| <i>dirname</i> //       | / or //     |
| <i>dirname</i> /a/b/    | /a          |
| <i>dirname</i> //a//b// | //a         |
| <i>dirname</i>          | Unspecified |
| <i>dirname</i> a        | . (\$? = 0) |
| <i>dirname</i> ""       | . (\$? = 0) |
| <i>dirname</i> /a       | /           |
| <i>dirname</i> /a/b     | /a          |
| <i>dirname</i> a/b      | a           |

13632 **RATIONALE**

13633 The *dirname* utility originated in System III. It has evolved through the System V releases to a  
 13634 version that matches the requirements specified in this description in System V Release 3. 4.3  
 13635 BSD and earlier versions did not include *dirname*.

13636 The behaviors of *basename* and *dirname* in this volume of IEEE Std. 1003.1-200x have been  
 13637 coordinated so that when *string* is a valid path name:

13638        `$(basename "string")`

13639        would be a valid file name for the file in the directory:

13640        `$(dirname "string")`

13641        This would not work for the versions of these utilities in early proposals due to the way  
13642        processing of trailing slashes was specified. Consideration was given to leaving processing  
13643        unspecified if there were trailing slashes, but this cannot be done; the System Interface  
13644        Definitions volume of IEEE Std. 1003.1-200x, Section 3.272, Path Name allows trailing slashes.  
13645        The *basename* and *dirname* utilities have to specify consistent handling for all valid path names.

13646 **FUTURE DIRECTIONS**

13647        None.

13648 **SEE ALSO**

13649        *basename*, Section 2.5 on page 43

13650 **CHANGE HISTORY**

13651        First released in Issue 2.

13652 **Issue 4**

13653        Aligned with the ISO/IEC 9945-2: 1993 standard.

13654 **NAME**

13655       du — estimate file space usage

13656 **SYNOPSIS**13657 UP       du [-a | -s][-kx][-H | -L][*file* ...]

13658

13659 **DESCRIPTION**

13660       By default, the *du* utility shall write to standard output the size of the file space allocated to, and  
 13661       the size of the file space allocated to each subdirectory of, the file hierarchy rooted in each of the  
 13662       specified files. By default, when a symbolic link is encountered on the command line or in the  
 13663       file hierarchy, *du* shall count the size of the symbolic link (rather than the file referenced by the  
 13664       link), and shall not follow the link to another portion of the file hierarchy. The size of the file  
 13665       space allocated to a file of type directory shall be defined as the sum total of space allocated to  
 13666       all files in the file hierarchy rooted in the directory plus the space allocated to the directory itself.

13667       When *du* cannot *stat()* files or *stat()* or read directories, it shall report an error condition and the  
 13668       final exit status is affected. Files with multiple links shall be counted and written for only one  
 13669       entry. The directory entry that is selected in the report is unspecified. By default, file sizes shall  
 13670       be written in 512-byte units, rounded up to the next 512-byte unit.

13671 **OPTIONS**

13672       The *du* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 13673       Section 12.2, Utility Syntax Guidelines.

13674       The following options shall be supported:

13675       **-a**       In addition to the default output, report the size of each file not of type directory in  
 13676       the file hierarchy rooted in the specified file. Regardless of the presence of the **-a**  
 13677       option, non-directories given as *file* operands shall always be listed.

13678       **-H**       If a symbolic link is specified on the command line, *du* shall count the size of the  
 13679       file or file hierarchy referenced by the link.

13680       **-k**       Write the files sizes in units of 1 024 bytes, rather than the default 512-byte units.

13681       **-L**       If a symbolic link is specified on the command line or encountered during the  
 13682       traversal of a file hierarchy, *du* shall count the size of the file or file hierarchy  
 13683       referenced by the link.

13684       **-s**       Instead of the default output, report only the total sum for each of the specified  
 13685       files.

13686       **-x**       When evaluating file sizes, evaluate only those files that have the same device as  
 13687       the file specified by the *file* operand.

13688       Specifying more than one of the mutually-exclusive options **-H** and **-L** shall not be considered  
 13689       an error. The last option specified shall determine the behavior of the utility.

13690 **OPERANDS**

13691       The following operand shall be supported:

13692       *file*       The path name of a file whose size is to be written. If no *file* is specified, the current  
 13693       directory shall be used.

13694 **STDIN**

13695       Not used.

13696 **INPUT FILES**

13697       None.

13698 **ENVIRONMENT VARIABLES**13699       The following environment variables shall affect the execution of *du*:

13700       **LANG**       Provide a default value for the internationalization variables that are unset or null.  
 13701                   If *LANG* is unset or null, the corresponding value from the implementation-  
 13702                   dependent default locale shall be used. If any of the internationalization variables  
 13703                   contains an invalid setting, the utility shall behave as if none of the variables had  
 13704                   been defined.

13705       **LC\_ALL**       If set to a non-empty string value, override the values of all the other  
 13706                   internationalization variables.

13707       **LC\_CTYPE**   Determine the locale for the interpretation of sequences of bytes of text data as  
 13708                   characters (for example, single-byte as opposed to multi-byte characters in  
 13709                   arguments).

13710       **LC\_MESSAGES**  
 13711                   Determine the locale that should be used to affect the format and contents of  
 13712                   diagnostic messages written to standard error.

13713 **XSI**       **NLSPATH**   Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

13714 **ASYNCHRONOUS EVENTS**

13715       Default.

13716 **STDOUT**

13717       The output from *du* shall consist of the amount of the space allocated to a file and the name of  
 13718       the file, in the following format:

13719       "%d %s\n", *<size>*, *<pathname>*

13720 **STDERR**

13721       Used only for diagnostic messages.

13722 **OUTPUT FILES**

13723       None.

13724 **EXTENDED DESCRIPTION**

13725       None.

13726 **EXIT STATUS**

13727       The following exit values shall be returned:

13728       0   Successful completion.

13729       >0   An error occurred.

13730 **CONSEQUENCES OF ERRORS**

13731       Default.

13732 **APPLICATION USAGE**

13733       Application writers should note that this utility need not be provided on systems that do not  
13734       support the User Portability Utilities option.

13735 **EXAMPLES**

13736       None.

13737 **RATIONALE**

13738       The use of 512-byte units is historical practice and maintains compatibility with *ls* and other  
13739       utilities in this volume of IEEE Std. 1003.1-200x. This does not mandate that the file system itself  
13740       be based on 512-byte blocks. The **-k** option was added as a compromise measure. It was agreed  
13741       by the standard developers that 512 bytes was the best default unit because of its complete  
13742       historical consistency on System V (*versus* the mixed 512/1 024-byte usage on BSD systems), and  
13743       that a **-k** option to switch to 1 024-byte units was a good compromise. Users who prefer the  
13744       1 024-byte quantity can easily alias *du* to *du -k* without breaking the many historical scripts  
13745       relying on the 512-byte units.

13746       The **-b** option was added to an early proposal to provide a resolution to the situation where  
13747       System V and BSD systems give figures for file sizes in *blocks*, which is an implementation-  
13748       dependent concept. (In common usage, the block size is 512 bytes for System V and 1 024 bytes  
13749       for BSD systems.) However, **-b** was later deleted, since the default was eventually decided as  
13750       512-byte units.

13751       Historical file systems provided no way to obtain exact figures for the space allocation given to  
13752       files. There are two known areas of inaccuracies in historical file systems: cases of *indirect blocks*  
13753       being used by the file system or *sparse* files yielding incorrectly high values. An indirect block is  
13754       space used by the file system in the storage of the file, but that need not be counted in the space  
13755       allocated to the file. A *sparse* file is one in which an *lseek()* call has been made to a position  
13756       beyond the end of the file and data has subsequently been written at that point. A file system  
13757       need not allocate all the intervening zero-filled blocks to such a file. It is up to the  
13758       implementation to define exactly how accurate its methods are.

13759       The **-a** and **-s** options were mutually-exclusive in the original version of *du*. The POSIX Shell  
13760       and Utilities description is implied by the language in the SVID where **-s** is described as causing  
13761       “only the grand total” to be reported. Some systems may produce output for **-sa**, but a Strictly  
13762       Conforming POSIX Shell and Utilities Application cannot use that combination.

13763       The **-a** and **-s** options were adopted from the SVID except that the System V behavior of not  
13764       listing non-directories explicitly given as operands, unless the **-a** option is specified, was  
13765       considered a bug; the BSD-based behavior (report for all operands) is mandated. The default  
13766       behavior of *du* in the SVID with regard to reporting the failure to read files (it produces no  
13767       messages) was considered counter-intuitive, and thus it was specified that the POSIX Shell and  
13768       Utilities default behavior shall be to produce such messages. These messages can be turned off  
13769       with shell redirection to achieve the System V behavior.

13770       The **-x** option is historical practice on recent BSD systems. It has been adopted by this volume of  
13771       IEEE Std. 1003.1-200x because there was no other historical method of limiting the *du* search to a  
13772       single file hierarchy. This limitation of the search is necessary to make it possible to obtain file  
13773       space usage information about a file system on which other file systems are mounted, without  
13774       having to resort to a lengthy *find* and *awk* script.

13775 **FUTURE DIRECTIONS**

13776       None.



13777 **SEE ALSO**13778 *ls*13779 **CHANGE HISTORY**

13780 First released in Issue 2.

13781 **Issue 4**

13782 Aligned with the ISO/IEC 9945-2: 1993 standard.

13783 **Issue 6**

13784 This utility is now marked as part of the User Portability Utilities option.

13785 The APPLICATION USAGE section is added.

13786 This utility is reinstated, as the LEGACY marking was incorrect in Issue 5.

13787 The obsolescent **-r** option has been removed.13788 The Open Group corrigenda item U025/3 has been applied. The *du* utility had incorrectly been  
13789 marked LEGACY.13790 The **-H** and **-L** options for symbolic links are added as described in the IEEE P1003.2b draft  
13791 standard.

13792 **NAME**

13793       echo — write arguments to standard output

13794 **SYNOPSIS**13795       echo [*string* ...]13796 **DESCRIPTION**

13797       The *echo* utility writes its arguments to standard output, followed by a <newline> character. If  
 13798       there are no arguments, only the <newline> character is written.

13799 **OPTIONS**

13800       The *echo* utility shall not recognize the "—" argument in the manner specified by Guideline 10  
 13801       of the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax  
 13802       Guidelines; "—" shall be recognized as a string operand.

13803 **MAN**       Implementations shall not support any options.13804 **OPERANDS**

13805       The following operands shall be supported:

13806 **MAN**       *string*       A string to be written to standard output. If any operand is **-n**, it shall be treated  
 13807       as a string, not an option. The following character sequences shall be recognized  
 13808       within any of the arguments:

13809        \a       Write an &lt;alert&gt; character.

13810       \b       Write a &lt;backspace&gt; character.

13811       \c       Suppress the <newline> character that otherwise follows the final  
 13812       argument in the output. All characters following the '\c' in the  
 13813       arguments shall be ignored.

13814       \f       Write a &lt;form-feed&gt; character.

13815       \n       Write a &lt;newline&gt; character.

13816       \r       Write a &lt;carriage-return&gt; character.

13817       \t       Write a &lt;tab&gt; character.

13818       \v       Write a &lt;vertical-tab&gt; character.

13819       \\       Write a backslash character.

13820       \0*num*   Write an 8-bit value that is the zero, one, two, or three-digit octal number  
 13821       *num*.

13822 **STDIN**

13823       Not used.

13824 **INPUT FILES**

13825       None.

13826 **ENVIRONMENT VARIABLES**13827       The following environment variables shall affect the execution of *echo*:

13828       *LANG*       Provide a default value for the internationalization variables that are unset or null.  
 13829       If *LANG* is unset or null, the corresponding value from the implementation-  
 13830       dependent default locale shall be used. If any of the internationalization variables  
 13831       contains an invalid setting, the utility shall behave as if none of the variables had  
 13832       been defined.

13833        **LC\_ALL**     If set to a non-empty string value, override the values of all the other  
 13834                   internationalization variables.

13835 MAN     **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 13836                   characters (for example, single-byte as opposed to multi-byte characters in  
 13837                   arguments).

13838        **LC\_MESSAGES**  
 13839                   Determine the locale that should be used to affect the format and contents of  
 13840                   diagnostic messages written to standard error.

13841 XSI     **NLSPATH**   Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

13842 **ASYNCHRONOUS EVENTS**  
 13843        Default.

13844 **STDOUT**  
 13845        The *echo* utility arguments shall be separated by single <space> characters and a <newline>  
 13846 MAN     character follows the last argument. Output transformations shall occur based on the escape  
 13847        sequences in the input. See the OPERANDS section.

13848 **STDERR**  
 13849        Used only for diagnostic messages.

13850 **OUTPUT FILES**  
 13851        None.

13852 **EXTENDED DESCRIPTION**  
 13853        None.

13854 **EXIT STATUS**  
 13855        The following exit values shall be returned:  
 13856        0    Successful completion.  
 13857        >0   An error occurred.

13858 **CONSEQUENCES OF ERRORS**  
 13859        Default.

13860 **APPLICATION USAGE**  
 13861        In the ISO/IEC 9945-2: 1993 standard, it was not possible to use *echo* portably across all systems  
 13862        that were not XSI-conformant unless both **-n** (as the first argument) and escape sequences were  
 13863        omitted.

13864        The *printf* utility can be used portably to emulate any of the traditional behaviors of the *echo*  
 13865        utility as follows:

13866        • The historic System V *echo* and the current requirements in this volume of  
 13867        IEEE Std. 1003.1-200x are equivalent to:

13868        `printf "%b\n" "$@"`

13869        • The BSD *echo* is equivalent to:

13870        `if [ "X$1" = "X-n" ]`  
 13871        `then`  
 13872        `shift`  
 13873        `printf "%s" "$@"`  
 13874        `else`  
 13875        `printf "%s\n" "$@"`

13876           fi

13877           New applications are encouraged to use *printf* instead of *echo*.

13878 **EXAMPLES**

13879           None.

13880 **RATIONALE**

13881           The *echo* utility has not been made obsolescent because of its extremely widespread use in  
13882           historical applications. Portable applications that wish to do prompting without <newline>s or  
13883           that could possibly be expecting to echo a **-n**, should use the new *printf* utility derived from the  
13884           Ninth Edition system.

13885           As specified, *echo* writes its arguments in the simplest of ways. The two different historical  
13886           versions of *echo* vary in fatally incompatible ways.

13887           The BSD *echo* checks the first argument for the string **-n** which causes it to suppress the  
13888           <newline> character that would otherwise follow the final argument in the output.

13889           The System V *echo* does not support any options, but allows escape sequences within its  
13890           operands, as described in the OPERANDS section.

13891           The *echo* utility does not support Utility Syntax Guideline 10 because historical applications  
13892           depend on *echo* to echo *all* of its arguments, except for the **-n** option in the BSD version.

13893 **FUTURE DIRECTIONS**

13894           None.

13895 **SEE ALSO**

13896           *printf*

13897 **CHANGE HISTORY**

13898           First released in Issue 2.

13899 **Issue 4**

13900           Aligned with the ISO/IEC 9945-2: 1993 standard.

13901 **Issue 5**

13902           In the OPTIONS section, the last sentence is changed to indicate that implementations “do not”  
13903           support any options; in the previous issue this said “need not”.

13904 **Issue 6**

13905           The following new requirements on POSIX implementations derive from alignment with the  
13906           Single UNIX Specification:

- 13907           • A set of character sequences is defined as *string* operands.
- 13908           • *LC\_CTYPE* is added to the list of environment variables affecting *echo*.
- 13909           • In the OPTIONS section, implementations shall not support any options.

13910 **NAME**

13911           ed — edit text

13912 **SYNOPSIS**13913           ed [-p *string*][-s][*file*]13914 **DESCRIPTION**

13915           The *ed* utility is a line-oriented text editor that uses two modes: *command mode* and *input mode*.  
 13916           In command mode the input characters shall be interpreted as commands, and in input mode  
 13917           they shall be interpreted as text. See the EXTENDED DESCRIPTION section.

13918 **OPTIONS**

13919           The *ed* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 13920           Section 12.2, Utility Syntax Guidelines.

13921           The following options shall be supported:

13922           **-B *string***    Use *string* as the prompt string when in command mode. By default, there shall be  
 13923                               no prompt string.

13924           **-s**            Suppress the writing of byte counts by **e**, **E**, **r**, and **w** commands and of the '!'  
 13925                               prompt after a *!command*.

13926 **OPERANDS**

13927           The following operand shall be supported:

13928           ***file***           If the *file* argument is given, *ed* shall simulate an **e** command on the file named by  
 13929                               the path name, *file*, before accepting commands from the standard input. If the *file*  
 13930                               operand is '-', the results are unspecified.

13931 **STDIN**

13932           The standard input shall be a text file consisting of commands, as described in the EXTENDED  
 13933           DESCRIPTION section.

13934 **INPUT FILES**

13935           The input files shall be text files.

13936 **ENVIRONMENT VARIABLES**

13937           The following environment variables shall affect the execution of *ed*:

13938           **HOME**           Determine the path name of the user's home directory.

13939           **LANG**           Provide a default value for the internationalization variables that are unset or null.  
 13940                               If *LANG* is unset or null, the corresponding value from the implementation-  
 13941                               dependent default locale shall be used. If any of the internationalization variables  
 13942                               contains an invalid setting, the utility shall behave as if none of the variables had  
 13943                               been defined.

13944           **LC\_ALL**          If set to a non-empty string value, override the values of all the other  
 13945                               internationalization variables.

13946           **LC\_COLLATE**

13947                               Determine the locale for the behavior of ranges, equivalence classes, and multi-  
 13948                               character collating elements within regular expressions.

13949           **LC\_CTYPE**       Determine the locale for the interpretation of sequences of bytes of text data as  
 13950                               characters (for example, single-byte as opposed to multi-byte characters in  
 13951                               arguments and input files) and the behavior of character classes within regular  
 13952                               expressions.

13953 *LC\_MESSAGES*

13954 Determine the locale that should be used to affect the format and contents of  
 13955 diagnostic messages written to standard error and informative messages written to  
 13956 standard output.

13957 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

13958 **ASYNCHRONOUS EVENTS**

13959 The *ed* utility shall take the standard action for all signals (see the ASYNCHRONOUS EVENTS  
 13960 section in Section 1.11 on page 25) with the following exceptions:

13961 SIGINT The *ed* utility shall interrupt its current activity, write the string "?\n" to standard  
 13962 output, and return to command mode (see the EXTENDED DESCRIPTION  
 13963 section).

13964 SIGHUP If the buffer is not empty and has changed since the last write, the *ed* utility shall  
 13965 attempt to write a copy of the buffer in a file. First, the file named **ed.hup** in the  
 13966 current directory shall be used; if that fails, the file named **ed.hup** in the directory  
 13967 named by the *HOME* environment variable shall be used. In any case, the *ed* utility  
 13968 shall exit without returning to command mode.

13969 SIGQUIT The *ed* utility shall ignore this event.

13970 **STDOUT**

13971 Various editing commands and the prompting feature (see **-p**) write to standard output, as  
 13972 described in the EXTENDED DESCRIPTION section.

13973 **STDERR**

13974 Used only for diagnostic messages.

13975 **OUTPUT FILES**

13976 The output files shall be text files whose formats are dependent on the editing commands given.

13977 **EXTENDED DESCRIPTION**

13978 The *ed* utility shall operate on a copy of the file it is editing; changes made to the copy shall have  
 13979 no effect on the file until a **w** (write) command is given. The copy of the text is called the *buffer*.

13980 Commands to *ed* have a simple and regular structure: zero, one, or two *addresses* followed by a  
 13981 single-character *command*, possibly followed by parameters to that command. These addresses  
 13982 specify one or more lines in the buffer. Every command that requires addresses has default  
 13983 addresses, so that the addresses very often can be omitted. If the **-p** option is specified, the  
 13984 prompt string shall be written to standard output before each command is read.

13985 In general, only one command can appear on a line. Certain commands allow text to be input.  
 13986 This text is placed in the appropriate place in the buffer. While *ed* is accepting text, it is said to be  
 13987 in *input mode*. In this mode, no commands shall be recognized; all input is merely collected.  
 13988 Input mode is terminated by entering a line consisting of two characters: a period ( **.** )  
 13989 followed by a <newline> character. This line is not considered part of the input text.

13990 **Regular Expressions in ed**

13991 The *ed* utility shall support basic regular expressions, as described in the System Interface  
 13992 Definitions volume of IEEE Std. 1003.1-200x, Section 9.3, Basic Regular Expressions. Since  
 13993 regular expressions in *ed* are always matched against single lines, never against any larger  
 13994 section of text, there is no way for a regular expression to match a <newline> character. A null  
 13995 RE shall be equivalent to the last RE encountered.

13996 Regular expressions are used in addresses to specify lines, and in some commands (for example,  
 13997 the **s** substitute command) to specify portions of a line to be substituted.

13998 **Addresses in ed**

13999 Addressing in *ed* relates to the current line. Generally, the current line is the last line affected by a  
 14000 command. The current line number is the address of the current line. If the edit buffer is not  
 14001 empty, the initial value for the current line shall be the last line in the edit buffer; otherwise, zero.

14002 Addresses shall be constructed as follows:

- 14003 1. The period character ( ' . ' ) shall address the current line.
- 14004 2. The dollar sign character ( ' \$ ' ) shall address the last line of the edit buffer.
- 14005 3. The positive decimal number *n* shall address the *n*th line of the edit buffer.
- 14006 4. The apostrophe-x character pair ( " ' x " ) shall address the line marked with the mark name  
 14007 character *x*, which shall be a lowercase letter from the portable character set. It shall be an  
 14008 error if the character has not been set to mark a line or if the line that was marked is not  
 14009 currently present in the edit buffer.
- 14010 5. A BRE enclosed by slash characters ( ' / ' ) shall address the first line found by searching  
 14011 forwards from the line following the current line toward the end of the edit buffer and  
 14012 stopping at the first line containing a string matching the BRE. The BRE consisting of a null  
 14013 BRE delimited by a pair of slash characters shall address the next line containing the last  
 14014 BRE encountered. In addition, the second slash can be omitted at the end of a command  
 14015 line. Within the BRE, a backslash-slash pair ( " \ / " ) shall represent a literal slash instead of  
 14016 the BRE delimiter. If necessary, the search shall wrap around to the beginning of the buffer  
 14017 and continue up to and including the current line, so that the entire buffer is searched.
- 14018 6. A BRE enclosed by question-mark characters ( ' ? ' ) shall address the first line found by  
 14019 searching backwards from the line preceding the current line toward the beginning of the  
 14020 edit buffer and stopping at the first line containing a string matching the BRE. The BRE  
 14021 consisting of a null BRE delimited by a pair of question-mark characters ( " ?? " ) shall  
 14022 address the previous line containing the last BRE encountered. In addition, the second  
 14023 question-mark can be omitted at the end of a command line. Within the BRE, a backslash-  
 14024 question-mark pair ( " \ ? " ) shall represent a literal question mark instead of the BRE  
 14025 delimiter. If necessary, the search shall wrap around to the end of the buffer and continue  
 14026 up to and including the current line, so that the entire buffer is searched.
- 14027 7. A plus-sign ( ' + ' ) or hyphen character ( ' - ' ) followed by a decimal number shall address  
 14028 the current line plus or minus the number. A plus-sign or hyphen character not followed  
 14029 by a decimal number shall address the current line plus or minus 1.

14030 Addresses can be followed by zero or more address offsets, optionally <blank>-separated.

14031 Address offsets are constructed as follows:

- 14032 • A plus-sign or hyphen character followed by a decimal number shall add or subtract,  
 14033 respectively, the indicated number of lines to or from the address. A plus-sign or hyphen  
 14034 character not followed by a decimal number shall add or subtract 1 to or from the address.
- 14035 • A decimal number shall add the indicated number of lines to the address.

14036 It shall not be an error for an intermediate address value to be less than zero or greater than the  
 14037 last line in the edit buffer. It shall be an error for the final address value to be less than zero or  
 14038 greater than the last line in the edit buffer. It shall be an error if a search for a BRE fails to find a  
 14039 matching line.

14040 Commands accept zero, one, or two addresses. If more than the required number of addresses  
 14041 are provided to a command that requires zero addresses, it shall be an error. Otherwise, if more  
 14042 than the required number of addresses are provided to a command, the addresses specified first

14043 shall be evaluated and then discarded until the maximum number of valid addresses remain, for  
14044 the specified command.

14045 Addresses shall be separated from each other by a comma (',' ) or semicolon character (';').  
14046 In the case of a semicolon separator, the current line ('.') shall be set to the first address, and  
14047 only then will the second address be calculated. This feature can be used to determine the  
14048 starting line for forwards and backwards searches; see rules 5. and 6.

14049 Addresses can be omitted on either side of the comma or semicolon separator, in which case the  
14050 resulting address pairs shall be as follows:

| Specified | Resulting   |
|-----------|-------------|
| ,         | 1 , \$      |
| , addr    | 1 ,a ddr    |
| addr ,    | addr , addr |
| ;         | . ; \$      |
| ; addr    | . ; addr    |
| addr ;    | addr ; addr |

14058 Any <blank> characters included between addresses, address separators, or address offsets shall  
14059 be ignored.

## 14060 **Commands in ed**

14061 In the following list of *ed* commands, the default addresses are shown in parentheses. The  
14062 number of addresses shown in the default shall be the number expected by the command. The  
14063 parentheses are not part of the address; they show that the given addresses are the default.

14064 It is generally invalid for more than one command to appear on a line. However, any command  
14065 (except **e**, **E**, **f**, **q**, **Q**, **r**, **w**, and **!**) can be suffixed by the letter **l**, **n**, or **p**; in which case, except for  
14066 the **l**, **n**, and **p** commands, the command shall be executed and then the new current line shall be  
14067 written as described below under the **l**, **n**, and **p** commands. When an **l**, **n**, or **p** suffix is used  
14068 with an **l**, **n**, or **p** command, the command shall write to standard output as described below, but  
14069 it is unspecified whether the suffix writes the current line again in the requested format or  
14070 whether the suffix has no effect. For example, the **pl** command (base **p** command with an **l**  
14071 suffix) shall either write just the current line or write it twice—once as specified for **p** and once  
14072 as specified for **l**. Also, the **g**, **G**, **v**, and **V** commands shall take a command as a parameter.

14073 Each address component can be preceded by zero or more <blank> characters. The command  
14074 letter can be preceded by zero or more <blank> characters. If a suffix letter (**l**, **n**, or **p**) is given,  
14075 the application shall ensure that it immediately follows the command.

14076 The **e**, **E**, **f**, **r**, and **w** commands shall take an optional *file* parameter, separated from the  
14077 command letter by one or more <blank> characters.

14078 If changes have been made in the buffer since the last **w** command that wrote the entire buffer,  
14079 *ed* shall warn the user if an attempt is made to destroy the editor buffer via the **e** or **q** commands.  
14080 The *ed* utility shall write the string:

14081 "?\n"

14082 (followed by an explanatory message if *help mode* has been enabled via the **H** command) to  
14083 standard output and shall continue in command mode with the current line number unchanged.  
14084 If the **e** or **q** command is repeated with no intervening command, it shall take effect.

14085 If a terminal disconnect is detected:



14086       • If the buffer is not empty and has changed since the last write, the *ed* utility shall attempt to  
 14087       write a copy of the buffer to a file named **ed.hup** in the current directory. If this write fails, *ed*  
 14088       shall attempt to write a copy of the buffer to a file name **ed.hup** in the directory named by the  
 14089       *HOME* environment variable. If both these attempts fail, *ed* shall exit without saving the  
 14090       buffer.

14091       • The *ed* utility shall not write the file to the currently remembered path name or return to  
 14092       command mode, and shall terminate with a non-zero exit status.

14093       If an end-of-file is detected on standard input:

14094       • If the *ed* utility is in input mode, *ed* shall terminate input mode and return to command mode.  
 14095       It is unspecified if any partially entered lines (that is, input text without a terminating  
 14096       <newline> character) are discarded from the input text.

14097       • If the *ed* utility is in command mode, it shall act as if a **q** command had been entered.

14098       If the closing delimiter of an RE or of a replacement string (for example, ' / ') in a **g**, **G**, **s**, **v**, or **V**  
 14099       command would be the last character before a <newline> character, that delimiter can be  
 14100       omitted, in which case the addressed line shall be written. For example, the following pairs of  
 14101       commands are equivalent:

```
14102 s/s1/s2 s/s1/s2/p
14103 g/s1 g/s1/p
14104 ?s1 ?s1?
```

14105       If an invalid command is entered, *ed* shall write the string:

14106       "?\\n"

14107       (followed by an explanatory message if *help mode* has been enabled via the **H** command) to  
 14108       standard output and shall continue in command mode with the current line number unchanged.

## 14109       Append Command

14110       *Synopsis:*       ( . )a  
 14111                       <text>  
 14112                       .

14113       The **a** command shall read the given text and append it after the addressed line; the current line  
 14114       number shall become the address of the last inserted line or, if there were none, the addressed  
 14115       line. Address 0 shall be valid for this command; it shall cause the appended text to be placed at  
 14116       the beginning of the buffer.

## 14117       Change Command

14118       *Synopsis:*       ( . , . )c  
 14119                       <text>  
 14120                       .

14121       The **c** command shall delete the addressed lines, then accept input text that replaces these lines;  
 14122       the current line shall be set to the address of the last line input; or, if there were none, at the line  
 14123       after the last line deleted; if the lines deleted were originally at the end of the buffer, the current  
 14124       line number shall be set to the address of the new last line; if no lines remain in the buffer, the  
 14125       current line number shall be set to zero. Address 0 shall be valid for this command; it shall be  
 14126       interpreted as if address 1 were specified.

**Delete Command**

*Synopsis:*     ( . , . )d

The **d** command shall delete the addressed lines from the buffer. The address of the line after the last line deleted shall become the current line number; if the lines deleted were originally at the end of the buffer, the current line number shall be set to the address of the new last line; if no lines remain in the buffer, the current line number shall be set to zero.

**Edit Command**

*Synopsis:*     e [*file*]

The **e** command shall delete the entire contents of the buffer and then read in the file named by the path name *file*. The current line number shall be set to the address of the last line of the buffer. If no path name is given, the currently remembered path name, if any, shall be used (see the **f** command). The number of bytes read shall be written to standard output, unless the **-s** option was specified, in the following format:

"%d\n", <number of bytes read>

The name *file* shall be remembered for possible use as a default path name in subsequent **e**, **E**, **r**, and **w** commands. If *file* is replaced by **'!'**, the rest of the line shall be taken to be a shell command line whose output is to be read. Such a shell command line shall not be remembered as the current *file*. All marks shall be discarded upon the completion of a successful **e** command. If the buffer has changed since the last time the entire buffer was written, the user shall be warned, as described previously.

**Edit Without Checking Command**

*Synopsis:*     E [*file*]

The **E** command shall possess all properties and restrictions of the **e** command except that the editor shall not check to see whether any changes have been made to the buffer since the last **w** command.

**File Name Command**

*Synopsis:*     f [*file*]

If *file* is given, the **f** command shall change the currently remembered path name to *file*; whether the name is changed or not, it shall then write the (possibly new) currently remembered path name to the standard output in the following format:

"%s\n", <pathname>

The current line number shall be unchanged.

**Global Command**

*Synopsis:*     (1,\$)g/*RE/command list*

In the **g** command, the first step shall be to mark every line that matches the given *RE*. Then, going sequentially from the beginning of the file to the end of the file, the given *command list* shall be executed for each marked line, with the current line number set to the address of that line. Any line modified by the *command list* shall be unmarked. When the **g** command completes, the current line number shall have the value assigned by the last command in the *command list*. If there were no matching lines, the current line number shall not be changed. A single command or the first of a list of commands shall appear on the same line as the global command. All lines

14168 of a multi-line list except the last line shall be ended with a backslash; the **a**, **i**, and **c** commands  
 14169 and associated input are permitted. The ' . ' terminating input mode can be omitted if it would  
 14170 be the last line of the *command list*. An empty *command list* shall be equivalent to the **p** command.  
 14171 The use of the **g**, **G**, **v**, **V**, and **!** commands in the *command list* produces undefined results. Any  
 14172 character other than <space> or <newline> can be used instead of a slash to delimit the *RE*.  
 14173 Within the *RE*, the *RE* delimiter itself can be used as a literal character if it is preceded by a  
 14174 backslash.

### 14175 Interactive Global Command

14176 *Synopsis:* ( 1 , \$ ) G / *RE* /

14177 In the **G** command, the first step shall be to mark every line that matches the given *RE*. Then,  
 14178 for every such line, that line shall be written, the current line number shall be set to the address  
 14179 of that line, and any one command (other than one of the **a**, **c**, **i**, **g**, **G**, **v**, and **V** commands) shall  
 14180 be read and executed. A <newline> character shall act as a null command (causing no action to  
 14181 be taken on the current line); an '&' shall cause the re-execution of the most recent non-null  
 14182 command executed within the current invocation of **G**. Note that the commands input as part  
 14183 of the execution of the **G** command can address and affect any lines in the buffer. The final value  
 14184 of the current line number shall be the value set by the last command successfully executed.  
 14185 (Note that the last command successfully executed shall be the **G** command itself if a command  
 14186 fails or the null command is specified.) If there were no matching lines, the current line number  
 14187 shall not be changed. The **G** command can be terminated by a SIGINT signal. Any character  
 14188 other than <space> or <newline> can be used instead of a slash to delimit the *RE* and the  
 14189 replacement. Within the *RE*, the *RE* delimiter itself can be used as a literal character if it is  
 14190 preceded by a backslash.

### 14191 Help Command

14192 *Synopsis:* h

14193 The **h** command shall write a short message to standard output that explains the reason for the  
 14194 most recent ' ? ' notification. The current line number shall be unchanged.

### 14195 Help-Mode Command

14196 *Synopsis:* H

14197 The **H** command shall cause *ed* to enter a mode in which help messages (see the **h** command)  
 14198 shall be written to standard output for all subsequent ' ? ' notifications. The **H** command  
 14199 alternatively shall turn this mode on and off; it is initially off. If the help-mode is being turned  
 14200 on, the **H** command also explains the previous ' ? ' notification, if there was one. The current  
 14201 line number shall be unchanged.

### 14202 Insert Command

14203 *Synopsis:* ( . ) i  
 14204 <text>  
 14205 .

14206 The **i** command shall insert the given text before the addressed line; the current line is set to the  
 14207 last inserted line or, if there was none, to the addressed line. This command differs from the **a**  
 14208 command only in the placement of the input text. Address 0 shall be valid for this command; it  
 14209 shall be interpreted as if address 1 were specified.

14210 **Join Command**14211 *Synopsis:* ( . , .+1 ) j

14212 The **j** command shall join contiguous lines by removing the appropriate <newline> characters. If  
 14213 exactly one address is given, this command shall do nothing. If lines are joined, the current line  
 14214 number shall be set to the address of the joined line; otherwise, the current line number shall be  
 14215 unchanged.

14216 **Mark Command**14217 *Synopsis:* ( . ) kx

14218 The **k** command shall mark the addressed line with name x, which the application shall ensure is  
 14219 a lowercase letter from the portable character set. The address " 'x'" shall then refer to this line;  
 14220 the current line number shall be unchanged.

14221 **List Command**14222 *Synopsis:* ( . , . ) l

14223 The **l** command shall write to standard output the addressed lines in a visually unambiguous  
 14224 form. The characters listed in the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 14225 Table 5-1, Escape Sequences and Associated Actions ( ' \ ' , ' \a ' , ' \b ' , ' \f ' , ' \r ' , ' \t ' ,  
 14226 ' \v ' ) shall be written as the corresponding escape sequence; the ' \n ' in that table is not  
 14227 applicable. Non-printable characters not in the table shall be written as one three-digit octal  
 14228 number (with a preceding backslash character) for each byte in the character (most significant  
 14229 byte first). If the size of a byte on the system is greater than nine bits, the format used for non-  
 14230 printable characters is implementation-dependent.

14231 Long lines shall be folded, with the point of folding indicated by writing backslash/<newline>  
 14232 character; the length at which folding occurs is unspecified, but should be appropriate for the  
 14233 output device. The end of each line shall be marked with a ' \$ ' , and ' \$ ' characters within the  
 14234 text shall be written with a preceding backslash. An **l** command can be appended to any other  
 14235 command other than **e**, **E**, **f**, **q**, **Q**, **r**, **w**, or **!**. The current line number shall be set to the address of  
 14236 the last line written.

14237 **Move Command**14238 *Synopsis:* ( . , . ) address

14239 The **m** command shall reposition the addressed lines after the line addressed by *address*.  
 14240 Address 0 shall be valid for *address* and cause the addressed lines to be moved to the beginning  
 14241 of the buffer. It shall be an error if address *address* falls within the range of moved lines. The  
 14242 current line number shall be set to the address of the last line moved.

14243 **Number Command**14244 *Synopsis:* ( . , . ) n

14245 The **n** command shall write to standard output the addressed lines, preceding each line by its  
 14246 line number and a <tab> character; the current line number shall be set to the address of the last  
 14247 line written. The **n** command can be appended to any command other than **e**, **E**, **f**, **q**, **Q**, **r**, **w**, or **!**.

14248 **Print Command**14249 *Synopsis:* ( . , . )p

14250 The **p** command shall write to standard output the addressed lines; the current line number shall  
 14251 be set to the address of the last line written. The **p** command can be appended to any command  
 14252 other than **e**, **E**, **f**, **q**, **Q**, **r**, **w**, or **!**.

14253 **Prompt Command**14254 *Synopsis:* P

14255 The **P** command shall cause *ed* to prompt with an asterisk ( **' \* '** ) (or *string*, if **-p** is specified) for  
 14256 all subsequent commands. The **P** command alternatively shall turn this mode on and off; it shall  
 14257 be initially on if the **-p** option is specified; otherwise, off. The current line number shall be  
 14258 unchanged.

14259 **Quit Command**14260 *Synopsis:* q

14261 The **q** command shall cause *ed* to exit. If the buffer has changed since the last time the entire  
 14262 buffer was written, the user shall be warned, as described previously.

14263 **Quit Without Checking Command**14264 *Synopsis:* Q

14265 The **Q** command shall cause *ed* to exit without checking whether changes have been made in the  
 14266 buffer since the last **w** command.

14267 **Read Command**14268 *Synopsis:* ( \$ )r [ *file* ]

14269 The **r** command shall read in the file named by the path name *file* and append it after the  
 14270 addressed line. If no *file* argument is given, the currently remembered path name, if any, shall be  
 14271 used (see the **e** and **f** commands). The currently remembered path name shall not be changed  
 14272 unless there is no remembered path name. Address 0 shall be valid for **r** and shall cause the file  
 14273 to be read at the beginning of the buffer. If the read is successful, and **-s** was not specified, the  
 14274 number of bytes read shall be written to standard output in the following format:

14275 "%d\n", &lt;number of bytes read&gt;

14276 The current line number shall be set to the address of the last line read in. If *file* is replaced by  
 14277 **' ! '**, the rest of the line shall be taken to be a shell command line whose output is to be read.  
 14278 Such a shell command line shall not be remembered as the current path name.

14279 **Substitute Command**14280 *Synopsis:* ( . , . )s/RE/replacement/flags

14281 The **s** command shall search each addressed line for an occurrence of the specified RE and  
 14282 replace either the first or all (non-overlapped) matched strings with the *replacement*; see the  
 14283 following description of the **g** suffix. It is an error if the substitution fails on every addressed  
 14284 line. Any character other than <space> or <newline> can be used instead of a slash to delimit the  
 14285 RE and the replacement. Within the RE, the RE delimiter itself can be used as a literal character if  
 14286 it is preceded by a backslash. The current line shall be set to the address of the last line on which  
 14287 a substitution occurred.

An ampersand ('&') appearing in the *replacement* shall be replaced by the string matching the RE on the current line. The special meaning of '&' in this context can be suppressed by preceding it by backslash. As a more general feature, the characters '\n', where *n* is a digit, shall be replaced by the text matched by the corresponding back-reference expression. When the character '%' is the only character in the *replacement*, the *replacement* used in the most recent substitute command shall be used as the *replacement* in the current substitute command; if there was no previous substitute command, the use of '%' in this manner shall be an error. The '%' shall lose its special meaning when it is in a replacement string of more than one character or is preceded by a backslash. For each backslash ('\') encountered in scanning *replacement* from beginning to end, the following character shall lose its special meaning (if any). It is unspecified what special meaning is given to any character other than '&', '\', '%', or digits.

A line can be split by substituting a <newline> character into it. The application shall ensure it escapes the <newline> character in the *replacement* by preceding it by backslash. Such substitution cannot be done as part of a **g** or **v** command list. The current line number shall be set to the address of the last line on which a substitution is performed. If no substitution is performed, the current line number shall be unchanged. If a line is split, a substitution shall be considered to have been performed on each of the new lines for the purpose of determining the new current line number. A substitution shall be considered to have been performed even if the replacement string is identical to the string that it replaces.

The application shall ensure that the value of *flags* is zero or more of:

**count**     Substitute for the *count*th occurrence only of the RE found on each addressed line.

**g**           Globally substitute for all non-overlapping instances of the RE rather than just the first one. If both **g** and *count* are specified, the results are unspecified.

**l**           Write to standard output the final line in which a substitution was made. The line shall be written in the format specified for the **l** command.

**n**           Write to standard output the final line in which a substitution was made. The line shall be written in the format specified for the **n** command.

**p**           Write to standard output the final line in which a substitution was made. The line shall be written in the format specified for the **p** command.

## Copy Command

*Synopsis:*     (.,.)*taddress*

The **t** command shall be equivalent to the **m** command, except that a copy of the addressed lines shall be placed after address *address* (which can be 0); the current line number shall be set to the address of the last line added.

## Undo Command

*Synopsis:*     u

The **u** command shall nullify the effect of the most recent command that modified anything in the buffer, namely the most recent **a**, **c**, **d**, **g**, **i**, **j**, **m**, **r**, **s**, **t**, **u**, **v**, **G**, or **V** command. All changes made to the buffer by a **g**, **G**, **v**, or **V** global command shall be undone as a single change; if no changes were made by the global command (such as with **g/RE/p**), the **u** command shall have no effect. The current line number shall be set to the value it had immediately before the command being undone started.

14330 **Global Non-Matched Command**14331 *Synopsis:* (1,\$)v/*RE/command list*14332 This command shall be equivalent to the global command **g** except that the lines that are marked  
14333 during the first step shall be those that do not match the *RE*.14334 **Interactive Global Not-Matched Command**14335 *Synopsis:* (1,\$)V/*RE/*14336 This command shall be equivalent to the interactive global command **G** except that the lines that  
14337 are marked during the first step shall be those that do not match the *RE*.14338 **Write Command**14339 *Synopsis:* (1,\$)w [*file*]14340 The **w** command shall write the addressed lines into the file named by the path name *file*. The  
14341 command shall create the file, if it does not exist, or shall replace the contents of the existing file.  
14342 The currently remembered path name shall not be changed unless there is no remembered path  
14343 name. If no path name is given, the currently remembered path name, if any, shall be used (see  
14344 the **e** and **f** commands); the current line number shall be unchanged. If the command is  
14345 successful, the number of bytes written shall be written to standard output, unless the **-s** option  
14346 was specified, in the following format:

14347 "%d\n", &lt;number of bytes written&gt;

14348 If *file* begins with '!', the rest of the line shall be taken to be a shell command line whose  
14349 standard input shall be the addressed lines. Such a shell command line shall not be remembered  
14350 as the current path name. This usage of the write command with '!' shall not be considered as  
14351 a "last **w** command that wrote the entire buffer", as described previously; thus, this alone shall  
14352 not prevent the warning to the user if an attempt is made to destroy the editor buffer via the **e** or  
14353 **q** commands.14354 **Line Number Command**14355 *Synopsis:* (\$)=14356 The line number of the addressed line shall be written to standard output in the following  
14357 format:

14358 "%d\n", &lt;line number&gt;

14359 The current line number shall be unchanged by this command.

14360 **Shell Escape Command**14361 *Synopsis:* !*command*14362 The remainder of the line after the '!' shall be sent to the command interpreter to be  
14363 interpreted as a shell command line. Within the text of that shell command line, the unescaped  
14364 character '%' shall be replaced with the remembered path name; if a '!' appears as the first  
14365 character of the command, it shall be replaced with the text of the previous shell command  
14366 executed via '!'. Thus, "!!" shall repeat the previous !*command*. If any replacements of '%' or  
14367 '!' are performed, the modified line shall be written to the standard output before *command* is  
14368 executed. The '!' command shall write:

14369 "!\n"

14370 to standard output upon completion, unless the **–s** option is specified. The current line number  
14371 shall be unchanged.

## 14372 **Null Command**

14373 *Synopsis:* ( . +1 )

14374 An address alone on a line shall cause the addressed line to be written. A <newline> character  
14375 alone shall be equivalent to "+1p". The current line number shall be set to the address of the  
14376 written line.

## 14377 **EXIT STATUS**

14378 The following exit values shall be returned:

14379 0 Successful completion without any file or command errors.

14380 >0 An error occurred.

## 14381 **CONSEQUENCES OF ERRORS**

14382 When an error in the input script is encountered, or when an error is detected that is a  
14383 consequence of the data (not) present in the file or due to an external condition such as a read or  
14384 write error:

14385 • If the standard input is a terminal device file, all input shall be flushed, and a new command  
14386 read.

14387 • If the standard input is a regular file, *ed* shall terminate with a non-zero exit status.

## 14388 **APPLICATION USAGE**

14389 Because of the extremely terse nature of the default error messages, the prudent script writer  
14390 begins the *ed* input commands with an **H** command, so that if any errors do occur at least some  
14391 clue as to the cause is made available.

14392 In previous versions, an obsolescent **–** option was described. This is no longer specified.  
14393 Applications should use the **–s** option. Using **–** as a file operand now produces unspecified  
14394 results. This allows implementations to continue to support the former required behavior.

## 14395 **EXAMPLES**

14396 None.

## 14397 **RATIONALE**

14398 The initial description of this utility was adapted from the SVID. It contains some features not  
14399 found in Version 7 or BSD-derived systems. Some of the differences between the POSIX and  
14400 BSD *ed* utilities include, but need not be limited to:

- 14401 • The BSD **–** option does not suppress the '!' prompt after a **!** command.
- 14402 • BSD does not support the special meanings of the '%' and '!' characters within a **!**  
14403 command.
- 14404 • BSD does not support the *addresses* ';' and ', '.
- 14405 • BSD allows the command/suffix pairs **pp**, **ll**, and so on, which are unspecified in this volume  
14406 of IEEE Std. 1003.1-200x.
- 14407 • BSD does not support the '!' character part of the **e**, **r**, or **w** commands.
- 14408 • A failed **g** command in BSD sets the line number to the last line searched if there are no  
14409 matches.
- 14410 • BSD does not default the *command list* to the **p** command.



- 14411 • BSD does not support the **G**, **h**, **H**, **n**, or **V** commands.
  - 14412 • On BSD, if there is no inserted text, the insert command changes the current line to the
  - 14413 referenced line -1; that is, the line before the specified line.
  - 14414 • On BSD, the *join* command with only a single address changes the current line to that
  - 14415 address.
  - 14416 • BSD does not support the **P** command; moreover, in BSD it is synonymous with the **p**
  - 14417 command.
  - 14418 • BSD does not support the *undo* of the commands **j**, **m**, **r**, **s**, or **t**.
  - 14419 • The Version 7 *ed* command **W**, and the BSD *ed* commands **W**, **wq**, and **z** are not present in this
  - 14420 volume of IEEE Std. 1003.1-200x.
- 14421 The **-s** option was added to allow the functionality of the now withdrawn **-** option in a manner
- 14422 compatible with the Utility Syntax Guidelines.
- 14423 In early proposals there was a limit, {ED\_FILE\_MAX}, that described the historical limitations of
- 14424 some *ed* utilities in their handling of large files; some of these have had problems with files larger
- 14425 than 100 000 bytes. It was this limitation that prompted much of the desire to include a *split*
- 14426 command in this volume of IEEE Std. 1003.1-200x. Since this limit was removed, this volume of
- 14427 IEEE Std. 1003.1-200x requires that implementations document the file size limits imposed by *ed*
- 14428 in the conformance document. The limit {ED\_LINE\_MAX} was also removed; therefore, the
- 14429 global limit {LINE\_MAX} is used for input and output lines.
- 14430 The manner in which the **l** command writes non-printable characters was changed to avoid the
- 14431 historical backspace-overstrike method. On video display terminals, the overstrike is ambiguous
- 14432 because most terminals simply replace overstruck characters, making the **l** format not useful for
- 14433 its intended purpose of unambiguously understanding the content of the line. The historical
- 14434 backslash escapes were also ambiguous. (The string "a\0011" could represent a line containing
- 14435 those six characters or a line containing the three characters 'a', a byte with a binary value of 1,
- 14436 and a 1.) In the format required here, a backslash appearing in the line is written as "\\ " so that
- 14437 the output is truly unambiguous. The method of marking the ends of lines was adopted from the
- 14438 *ex* editor and is required for any line ending in <space>s; the '\$' is placed on all lines so that a
- 14439 real '\$' at the end of a line cannot be misinterpreted.
- 14440 Systems with bytes too large to fit into three octal digits must devise other means of displaying
- 14441 non-printable characters. Consideration was given to requiring that the number of octal digits be
- 14442 large enough to hold a byte, but this seemed to be too confusing for applications on the vast
- 14443 majority of systems where three digits are adequate. It would be theoretically possible for the
- 14444 application to use the *getconf* utility to find out the CHAR\_BIT value and deal with such an
- 14445 algorithm; however, there is really no portable way that an application can use the octal values
- 14446 of the bytes across various coded character sets, so the additional specification was not
- 14447 worthwhile.
- 14448 The description of how a NUL is written was removed. The NUL character cannot be in text
- 14449 files, and this volume of IEEE Std. 1003.1-200x should not dictate behavior in the case of
- 14450 undefined, erroneous input.
- 14451 Unlike some of the other editing utilities, the file names accepted by the **E**, **e**, **R**, and **r** commands
- 14452 are not patterns.
- 14453 Early proposals stated that the **-p** option worked only when standard input was associated with
- 14454 a terminal device. This has been changed to conform to historical implementations, thereby
- 14455 allowing applications to interpose themselves between a user and the *ed* utility.

14456 The form of the substitute command that uses the **n** suffix was limited in some historical  
 14457 documentation (where this was described incorrectly as “backreferencing”). This limit has been  
 14458 omitted because there is no reason an editor processing lines of {LINE\_MAX} length should have  
 14459 this restriction. The command **s/x/X/2 047** should be able to substitute the 2 047th occurrence of **x**  
 14460 on a line.

14461 The use of printing commands with printing suffixes (such as **pn**, **lp**, and so on) was made  
 14462 unspecified because BSD-based systems allow this, whereas System V does not.

14463 Some BSD-based systems exit immediately upon receipt of end-of-file if all of the lines in the file  
 14464 have been deleted. Since this volume of IEEE Std. 1003.1-200x refers to the **q** command in this  
 14465 instance, such behavior is not allowed.

14466 Some historical implementations returned exit status zero even if command errors had occurred;  
 14467 this is not allowed by this volume of IEEE Std. 1003.1-200x.

14468 Some historical implementations contained a bug that allowed a single period to be entered in  
 14469 input mode as <backslash> <period> <newline>. This is not allowed by the *ed* because there is  
 14470 no description of escaping any of the characters in input mode; backslashes are entered into the  
 14471 buffer exactly as typed. The typical method of entering a single period has been to precede it  
 14472 with another character and then use the substitute command to delete that character.

14473 It is difficult under some modes of some versions of historical operating system terminal drivers  
 14474 to distinguish between an end-of-file condition and terminal disconnect. The ISO POSIX-2  
 14475 standard does not require implementations to distinguish between the two situations, which  
 14476 permits historical implementations of the *ed* utility on historical platforms to conform.  
 14477 Implementations are encouraged to distinguish between the two, if possible, and take  
 14478 appropriate action on terminal disconnect.

14479 Historically, *ed* accepted a zero address for the **a** and **r** commands in order to insert text at the  
 14480 start of the edit buffer. When the buffer was empty the command **.=** returned zero.  
 14481 IEEE Std. 1003.1-200x requires conformance to historical practice.

14482 For consistency with the **a** and **r** commands and better user functionality, the **i** and **c** commands  
 14483 must also accept an address of 0, in which case **0i** is treated as **1i** and likewise for the **c**  
 14484 command.

14485 All of the following are valid addresses:

14486 **+++** Three lines after the current line.

14487 **/pattern/-** One line before the next occurrence of pattern.

14488 **-2** Two lines before the current line.

14489 **3 ——— 2** Line one (note the intermediate negative address).

14490 **1 2 3** Line six.

14491 Any number of addresses can be provided to commands taking addresses; for example,  
 14492 **"1,2,3,4,5p"** prints lines 4 and 5, because two is the greatest valid number of addresses  
 14493 accepted by the **print** command. This, in combination with the semicolon delimiter, permits  
 14494 users to create commands based on ordered patterns in the file. For example, the command  
 14495 **"3;/foo/;+2p"** will display the first line after line 3 that contains the pattern *foo*, plus the next  
 14496 two lines. Note that the address **"3;"** must still be evaluated before being discarded, because  
 14497 the search origin for the **"/foo/"** command depends on this.

14498 Historically, *ed* disallowed address chains, as discussed above, consisting solely of comma or  
 14499 semicolon separators; for example, **" , , , "** or **" ; ; ; "** were considered an error. For consistency of  
 14500 address specification, this restriction is removed. The following table lists some of the address

14501 forms now possible:

| Address | Addr1 | Addr2 | Status     | Comment               |
|---------|-------|-------|------------|-----------------------|
| 7,      | 7     | 7     | Historical | Valid, but erroneous. |
| 7,5,    | 5     | 5     | Historical |                       |
| 7,5,9   | 5     | 9     | Historical |                       |
| 7,9     | 7     | 9     | Historical |                       |
| 7,+     | 7     | 8     | Historical |                       |
| ,       | 1     | \$    | Historical |                       |
| ,7      | 1     | 7     | Extension  |                       |
| ,,      | \$    | \$    | Extension  |                       |
| ,;      | \$    | \$    | Extension  |                       |
| 7;      | 7     | 7     | Historical |                       |
| 7;5;    | 5     | 5     | Historical |                       |
| 7;5;9   | 5     | 9     | Historical |                       |
| 7;5,9   | 5     | 9     | Historical |                       |
| 7;\$;4  | \$    | 4     | Historical |                       |
| 7;9     | 7     | 9     | Historical |                       |
| 7;+     | 7     | 8     | Historical |                       |
| ;       | .     | \$    | Historical |                       |
| ;7      | .     | 7     | Extension  |                       |
| ;;      | \$    | \$    | Extension  |                       |
| ;,      | \$    | \$    | Extension  |                       |

14523 Historically, values could be added to addresses by including them after one or more <blank>  
 14524 characters; for example, "3 - 5p" wrote the seventh line of the file, and "/foo/ 5" was the  
 14525 same as "5 /foo/". However, only absolute values could be added; for example, "5 /foo/"  
 14526 was an error. IEEE Std. 1003.1-200x requires conformance to historical practice.

14527 Historically, *ed* accepted the '^' character as an address, in which case it was identical to the  
 14528 hyphen character. IEEE Std. 1003.1-200x does not require or prohibit this behavior.

#### 14529 FUTURE DIRECTIONS

14530 None.

#### 14531 SEE ALSO

14532 *ex, sed, sh, vi*

#### 14533 CHANGE HISTORY

14534 First released in Issue 2.

#### 14535 Issue 4

14536 Aligned with the ISO/IEC 9945-2:1993 standard.

#### 14537 Issue 5

14538 In the OPTIONS section, the meaning of -s and - is clarified.

14539 Second FUTURE DIRECTION added.

#### 14540 Issue 6

14541 The obsolescent single-minus form has been removed.

14542 A second APPLICATION USAGE note has been added.

14543 The Open Group corrigenda item U025/2 has been applied, correcting the description of the Edit  
 14544 section.

|       |                                                                                                                                                                                                                                                                               |  |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| 14545 | The <i>ed</i> utility is updated to align with the IEEE P1003.2b draft standard. This includes addition of the treatment of the SIGQUIT signal, changes to <i>ed</i> addressing, changes to processing when end-of-file is detected and when terminal disconnect is detected. |  |
| 14546 |                                                                                                                                                                                                                                                                               |  |
| 14547 |                                                                                                                                                                                                                                                                               |  |
| 14548 | The normative text is reworded to avoid use of the term “must” for application requirements.                                                                                                                                                                                  |  |

14549 **NAME**14550 `env` — set the environment for command invocation14551 **SYNOPSIS**14552 `env [-i][name=value]... [utility [argument...]]`14553 **DESCRIPTION**14554 The *env* utility shall obtain the current environment, modify it according to its arguments, then  
14555 invoke the utility named by the *utility* operand with the modified environment.14556 Optional arguments shall be passed to *utility*.14557 If no *utility* operand is specified, the resulting environment shall be written to the standard  
14558 output, with one *name=value* pair per line.14559 **OPTIONS**14560 The *env* utility shall conform to the System Interface Definitions volume of  
14561 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

14562 The following options shall be supported:

14563 **-i** Invoke *utility* with exactly the environment specified by the arguments; the  
14564 inherited environment shall be ignored completely.14565 **OPERANDS**

14566 The following operands shall be supported:

14567 **name=value** Arguments of the form *name=value* shall modify the execution environment, and  
14568 shall be placed into the inherited environment before the *utility* is invoked.14569 **utility** The name of the utility to be invoked. If the *utility* operand names any of the  
14570 special built-in utilities in Section 2.14 on page 96, the results are undefined.14571 **argument** A string to pass as an argument for the invoked utility.14572 **STDIN**

14573 Not used.

14574 **INPUT FILES**

14575 None.

14576 **ENVIRONMENT VARIABLES**14577 The following environment variables shall affect the execution of *env*:14578 **LANG** Provide a default value for the internationalization variables that are unset or null.  
14579 If *LANG* is unset or null, the corresponding value from the implementation-  
14580 dependent default locale shall be used. If any of the internationalization variables  
14581 contains an invalid setting, the utility shall behave as if none of the variables had  
14582 been defined.14583 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
14584 internationalization variables.14585 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
14586 characters (for example, single-byte as opposed to multi-byte characters in  
14587 arguments).14588 **LC\_MESSAGES**14589 Determine the locale that should be used to affect the format and contents of  
14590 diagnostic messages written to standard error.

14591 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.  
 14592 **PATH** Determine the location of the *utility*, as described in the System Interface  
 14593 Definitions volume of IEEE Std. 1003.1-200x, Chapter 8, Environment Variables. If  
 14594 *PATH* is specified as a *name=value* operand to *env*, the *value* given shall be used in  
 14595 the search for *utility*.

#### 14596 ASYNCHRONOUS EVENTS

14597 Default.

#### 14598 STDOUT

14599 If no *utility* operand is specified, each *name=value* pair in the resulting environment shall be  
 14600 written in the form:

14601 "%s=%s\n", <name>, <value>

14602 If the *utility* operand is specified, the *env* utility shall not write to standard output.

#### 14603 STDERR

14604 Used only for diagnostic messages.

#### 14605 OUTPUT FILES

14606 None.

#### 14607 EXTENDED DESCRIPTION

14608 None.

#### 14609 EXIT STATUS

14610 If the *utility* utility is invoked, the exit status of *env* shall be the exit status of *utility*; otherwise,  
 14611 the *env* utility shall exit with one of the following values:

14612 0 The *env* utility completed successfully.

14613 1–125 An error occurred in the *env* utility.

14614 126 The utility specified by *utility* was found but could not be invoked.

14615 127 The utility specified by *utility* could not be found.

#### 14616 CONSEQUENCES OF ERRORS

14617 Default.

#### 14618 APPLICATION USAGE

14619 The *command*, *env*, *nice*, *nohup*, *time*, and *xargs* utilities have been specified to use exit code 127 if  
 14620 an error occurs so that applications can distinguish “failure to find a utility” from “invoked  
 14621 utility exited with an error indication”. The value 127 was chosen because it is not commonly  
 14622 used for other meanings; most utilities use small values for “normal error conditions” and the  
 14623 values above 128 can be confused with termination due to receipt of a signal. The value 126 was  
 14624 chosen in a similar manner to indicate that the utility could be found, but not invoked. Some  
 14625 scripts produce meaningful error messages differentiating the 126 and 127 cases. The distinction  
 14626 between exit codes 126 and 127 is based on KornShell practice that uses 127 when all attempts to  
 14627 *exec* the utility fail with [ENOENT], and uses 126 when any attempt to *exec* the utility fails for  
 14628 any other reason.

14629 Historical implementations of the *env* utility use the *execvp()* or *execlp()* functions defined in the  
 14630 System Interfaces volume of IEEE Std. 1003.1-200x to invoke the specified utility; this provides  
 14631 better performance and keeps users from having to escape characters with special meaning to  
 14632 the shell. Therefore, shell functions, special built-ins, and built-ins that are only provided by the  
 14633 shell are not found.

**14634 EXAMPLES**

14635       The following command:

14636       `env -i PATH=/mybin mygrep xyz myfile`

14637       invokes the command *mygrep* with a new *PATH* value as the only entry in its environment. In  
14638       this case, *PATH* is used to locate *mygrep*, which then must reside in **/mybin**.

**14639 RATIONALE**

14640       As with all other utilities that invoke other utilities, this volume of IEEE Std. 1003.1-200x only  
14641       specifies what *env* does with standard input, standard output, standard error, input files, and  
14642       output files. If a utility is executed, it is not constrained by the specification of input and output  
14643       by *env*.

14644       The **-i** option was added to allow the functionality of the withdrawn **-** option in a manner  
14645       compatible with the Utility Syntax Guidelines.

14646       Some have suggested that *env* is redundant since the same effect is achieved by:

14647       `name=value ... utility [ argument ... ]`

14648       The example is equivalent to *env* when an environment variable is being added to the  
14649       environment of the command, but not when the environment is being set to the given value.

14650       The *env* utility also writes out the current environment if invoked without arguments. There is  
14651       sufficient functionality beyond what the example provides to justify inclusion of *env*.

**14652 FUTURE DIRECTIONS**

14653       None.

**14654 SEE ALSO**

14655       Section 2.5 on page 43

**14656 CHANGE HISTORY**

14657       First released in Issue 2.

**14658 Issue 4**

14659       Aligned with the ISO/IEC 9945-2:1993 standard.

## 14660 NAME

14661 ex — text editor

14662 **Notes to Reviewers**14663 *This section with side shading will not appear in the final copy. - Ed.*

14664 This page has undergone significant revision due to the 1003.2b merger. The following D1 XCU  
 14665 ERNs should be reviewed now the merge is complete: 234-239, 241-245.

## 14666 SYNOPSIS

14667 UP `ex [-rR][-l][-s | -v][-c command]-t tagstring][-w size][file ...]`

14668

## 14669 DESCRIPTION

14670 The **ex** utility is a line-oriented text editor. There are two other modes of the editor—open and  
 14671 visual—in which screen-oriented editing is available. This is described more fully by the **ex open**  
 14672 and **visual** commands and in *vi*.

14673 This section uses the term *edit buffer* to describe the current working text. No specific  
 14674 implementation is implied by this term. All editing changes are performed on the edit buffer,  
 14675 and no changes to it shall affect any file until an editor command writes the file.

14676 Certain terminals do not have all the capabilities necessary to support the complete **ex** definition,  
 14677 such as the full-screen editing commands (*visual mode* or *open mode*). When these commands  
 14678 cannot be supported on such terminals, this condition shall not produce an error message such  
 14679 as “not an editor command” or report a syntax error. The implementation may either accept the  
 14680 commands and produce results on the screen that are the result of an unsuccessful attempt to  
 14681 meet the requirements of this volume of IEEE Std. 1003.1-200x or report an error describing the  
 14682 terminal-related deficiency.

## 14683 OPTIONS

14684 The **ex** utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 14685 Section 12.2, Utility Syntax Guidelines.

14686 The following options shall be supported:

14687 **-c command** Specify an initial command to be executed in the first edit buffer loaded from an  
 14688 existing file (see the EXTENDED DESCRIPTION section). Implementations may  
 14689 support more than a single **-c** option. In such implementations, the specified  
 14690 commands shall be executed in the order specified on the command line.

14691 MAN **-l** (The letter ell.) Set lisp mode; indents appropriately for Lisp code; the **O**, **f**, **[[**, and  
 14692 **]]** commands in visual mode are modified to have meaning for Lisp.

14693 **-r** Recover the named files (see the EXTENDED DESCRIPTION section). Recovery  
 14694 information for a file shall be saved during an editor or system crash (for example,  
 14695 when the editor is terminated by a signal which the editor can catch), or after the  
 14696 use of an **ex preserve** command.

14697 A *crash* in this context is an unexpected failure of the system or utility that requires  
 14698 restarting the failed system or utility. A system crash implies that any utilities  
 14699 running at the time also crash. In the case of an editor or system crash, the number  
 14700 of changes to the edit buffer (since the most recent **preserve** command) that will be  
 14701 recovered is unspecified.

14702 If no *file* operands are given and the **-t** option is not specified, all other options, the  
 14703 *EXINIT* variable, and any **.exrc** files shall be ignored; a list of all recoverable files  
 14704 available to the invoking user shall be written, and the editor shall exit normally



|       |                              |                                                                                                                |  |
|-------|------------------------------|----------------------------------------------------------------------------------------------------------------|--|
| 14705 |                              | without further action.                                                                                        |  |
| 14706 | <b>-R</b>                    | Set <b>readonly</b> edit option.                                                                               |  |
| 14707 | <b>-s</b>                    | Prepare <i>ex</i> for batch use by taking the following actions:                                               |  |
| 14708 |                              | • Suppress writing prompts and informational (but not diagnostic) messages.                                    |  |
| 14709 |                              | • Ignore the value of <i>TERM</i> and any implementation default terminal type and                             |  |
| 14710 |                              | assume the terminal is a type incapable of supporting open or visual modes;                                    |  |
| 14711 |                              | see the <b>visual</b> command and the description of <i>vi</i> .                                               |  |
| 14712 |                              | • Suppress the use of the <i>EXINIT</i> environment variable and the reading of any                            |  |
| 14713 |                              | <b>.exrc</b> file; see the EXTENDED DESCRIPTION section.                                                       |  |
| 14714 |                              | • Suppress autoindentation, ignoring the value of the <b>autoindent</b> edit option.                           |  |
| 14715 | <b>-t tagstring</b>          | Edit the file containing the specified <i>tagstring</i> ; see <i>ctags</i> . The tags feature                  |  |
| 14716 |                              | represented by <b>-t tagstring</b> and the <b>tag</b> command is optional. It shall be provided                |  |
| 14717 |                              | on any system that also provides a conforming implementation of <i>ctags</i> ; otherwise,                      |  |
| 14718 |                              | the use of <b>-t</b> produces undefined results. On any system, it shall be an error to                        |  |
| 14719 |                              | specify more than a single <b>-t</b> option.                                                                   |  |
| 14720 | <b>-v</b>                    | Begin in visual mode (see <i>vi</i> ).                                                                         |  |
| 14721 | <b>-w size</b>               | Set the value of the <i>window</i> editor option to <i>size</i> .                                              |  |
| 14722 | <b>OPERANDS</b>              |                                                                                                                |  |
| 14723 |                              | The following operand shall be supported:                                                                      |  |
| 14724 | <i>file</i>                  | A path name of a file to be edited.                                                                            |  |
| 14725 | <b>STDIN</b>                 |                                                                                                                |  |
| 14726 |                              | The standard input consists of a series of commands and input text, as described in the                        |  |
| 14727 |                              | EXTENDED DESCRIPTION section. The implementation may limit each line of standard input                         |  |
| 14728 |                              | to a length of {LINE_MAX}.                                                                                     |  |
| 14729 |                              | If the standard input is not a terminal device, it shall be as if the <b>-s</b> option had been specified.     |  |
| 14730 |                              | If a read from the standard input returns an error, or if the editor detects an end-of-file condition          |  |
| 14731 |                              | from the standard input, it shall be equivalent to a SIGHUP asynchronous event.                                |  |
| 14732 | <b>INPUT FILES</b>           |                                                                                                                |  |
| 14733 |                              | Input files shall be text files or files that would be text files except for an incomplete last line that      |  |
| 14734 |                              | is not longer than {LINE_MAX}-1 bytes in length and contains no NUL characters. By default,                    |  |
| 14735 |                              | any incomplete last line shall be treated as if it had a trailing <newline> character. The editing of          |  |
| 14736 |                              | other forms of files may optionally be allowed by <i>ex</i> implementations.                                   |  |
| 14737 |                              | The <b>.exrc</b> files and source files shall be text files consisting of <i>ex</i> commands; see the EXTENDED |  |
| 14738 |                              | DESCRIPTION section.                                                                                           |  |
| 14739 |                              | By default, the editor shall read lines from the files to be edited without interpreting any of those          |  |
| 14740 |                              | lines as any form of editor command.                                                                           |  |
| 14741 | <b>ENVIRONMENT VARIABLES</b> |                                                                                                                |  |
| 14742 |                              | The following environment variables shall affect the execution of <i>ex</i> :                                  |  |
| 14743 | <i>COLUMNS</i>               | Override the system-selected horizontal screen size. See the System Interface                                  |  |
| 14744 |                              | Definitions volume of IEEE Std. 1003.1-200x, Chapter 8, Environment Variables for                              |  |
| 14745 |                              | valid values and results when it is unset or null.                                                             |  |

|       |                                                                                    |                                                                                                                         |
|-------|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------|
| 14746 | <i>EXINIT</i>                                                                      | Determine a list of <i>ex</i> commands that are executed on editor start-up. See the                                    |
| 14747 |                                                                                    | EXTENDED DESCRIPTION section for more details of the initialization phase.                                              |
| 14748 | <i>HOME</i>                                                                        | Determine a path name of a directory that shall be searched for an editor start-up                                      |
| 14749 |                                                                                    | file named <i>.exrc</i> ; see the EXTENDED DESCRIPTION section.                                                         |
| 14750 | <i>LANG</i>                                                                        | Provide a default value for the internationalization variables that are unset or null.                                  |
| 14751 |                                                                                    | If <i>LANG</i> is unset or null, the corresponding value from the implementation-                                       |
| 14752 |                                                                                    | dependent default locale shall be used. If any of the internationalization variables                                    |
| 14753 |                                                                                    | contains an invalid setting, the utility shall behave as if none of the variables had                                   |
| 14754 |                                                                                    | been defined.                                                                                                           |
| 14755 | <i>LC_ALL</i>                                                                      | If set to a non-empty string value, override the values of all the other                                                |
| 14756 |                                                                                    | internationalization variables.                                                                                         |
| 14757 | <i>LC_COLLATE</i>                                                                  |                                                                                                                         |
| 14758 |                                                                                    | Determine the locale for the behavior of ranges, equivalence classes, and multi-                                        |
| 14759 |                                                                                    | character collating elements within regular expressions.                                                                |
| 14760 | <i>LC_CTYPE</i>                                                                    | Determine the locale for the interpretation of sequences of bytes of text data as                                       |
| 14761 |                                                                                    | characters (for example, single-byte as opposed to multi-byte characters in                                             |
| 14762 |                                                                                    | arguments and input files), the behavior of character classes within regular                                            |
| 14763 |                                                                                    | expressions, the classification of characters as uppercase or lowercase letters, the                                    |
| 14764 |                                                                                    | case conversion of letters, and the detection of word boundaries.                                                       |
| 14765 | <i>LC_MESSAGES</i>                                                                 |                                                                                                                         |
| 14766 |                                                                                    | Determine the locale that should be used to affect the format and contents of                                           |
| 14767 |                                                                                    | diagnostic messages written to standard error.                                                                          |
| 14768 | <i>LINES</i>                                                                       | Override the system-selected vertical screen size, used as the number of lines in a                                     |
| 14769 |                                                                                    | screenful and the vertical screen size in visual mode. See the System Interface                                         |
| 14770 |                                                                                    | Definitions volume of IEEE Std. 1003.1-200x, Chapter 8, Environment Variables for                                       |
| 14771 |                                                                                    | valid values and results when it is unset or null.                                                                      |
| 14772 | XSI <i>NLSPATH</i>                                                                 | Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .                                   |
| 14773 | <i>PATH</i>                                                                        | Determine the search path for the shell command specified in the <i>ex</i> editor                                       |
| 14774 |                                                                                    | commands <i>!</i> , <i>shell</i> , <i>read</i> , and <i>write</i> , and the open and visual mode command <i>!</i> ; see |
| 14775 |                                                                                    | the description of command search and execution in Section 2.9.1.1 on page 69.                                          |
| 14776 | <i>SHELL</i>                                                                       | Determine the preferred command line interpreter for use as the default value of                                        |
| 14777 |                                                                                    | the <i>shell</i> edit option.                                                                                           |
| 14778 | <i>TERM</i>                                                                        | Determine the name of the terminal type. If this variable is unset or null, an                                          |
| 14779 |                                                                                    | unspecified default terminal type shall be used.                                                                        |
| 14780 | <b>ASYNCHRONOUS EVENTS</b>                                                         |                                                                                                                         |
| 14781 | The following symbol is used in this and following sections to specify command and |                                                                                                                         |
| 14782 | asynchronous event actions:                                                        |                                                                                                                         |
| 14783 | <i>complete write</i>                                                              |                                                                                                                         |
| 14784 |                                                                                    | A complete write is a write of the entire contents of the edit buffer to a file of a type                               |
| 14785 |                                                                                    | other than a terminal device, or the saving of the edit buffer caused by the user                                       |
| 14786 |                                                                                    | executing the <i>ex preserve</i> command. Writing the contents of the edit buffer to a                                  |
| 14787 |                                                                                    | temporary file that will be removed when the editor exits shall not be considered a                                     |
| 14788 |                                                                                    | complete write.                                                                                                         |
| 14789 | The following actions shall be taken upon receipt of signals:                      |                                                                                                                         |

|       |                             |                                                                                                                                                                                                                                                                                                                                                         |
|-------|-----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 14790 | SIGINT                      | If the standard input is not a terminal device, <i>ex</i> shall not write the file or return to command or text input mode, and shall exit with a non-zero exit status.                                                                                                                                                                                 |
| 14791 |                             |                                                                                                                                                                                                                                                                                                                                                         |
| 14792 |                             | Otherwise, if executing an open or visual text input mode command, <i>ex</i> in receipt of SIGINT shall behave identically to its receipt of the <ESC> character.                                                                                                                                                                                       |
| 14793 |                             |                                                                                                                                                                                                                                                                                                                                                         |
| 14794 |                             | Otherwise:                                                                                                                                                                                                                                                                                                                                              |
| 14795 |                             | 1. If executing an <i>ex</i> text input mode command, all input lines that have been completely entered shall be resolved into the edit buffer, and any partially entered line shall be discarded.                                                                                                                                                      |
| 14796 |                             |                                                                                                                                                                                                                                                                                                                                                         |
| 14797 |                             |                                                                                                                                                                                                                                                                                                                                                         |
| 14798 |                             | 2. If there is a currently executing command, it shall be aborted and a message displayed. Unless otherwise specified by the <i>ex</i> or <i>vi</i> command descriptions, it is unspecified whether any lines modified by the executing command appear modified, or as they were before being modified by the executing command, in the buffer.         |
| 14799 |                             |                                                                                                                                                                                                                                                                                                                                                         |
| 14800 |                             |                                                                                                                                                                                                                                                                                                                                                         |
| 14801 |                             |                                                                                                                                                                                                                                                                                                                                                         |
| 14802 |                             |                                                                                                                                                                                                                                                                                                                                                         |
| 14803 |                             | If the currently executing command was a motion command, its associated command shall be discarded.                                                                                                                                                                                                                                                     |
| 14804 |                             |                                                                                                                                                                                                                                                                                                                                                         |
| 14805 |                             | 3. If in open or visual command mode, the terminal shall be alerted.                                                                                                                                                                                                                                                                                    |
| 14806 |                             | 4. The editor shall then return to command mode.                                                                                                                                                                                                                                                                                                        |
| 14807 | SIGCONT                     | The screen shall be refreshed if in open or visual mode.                                                                                                                                                                                                                                                                                                |
| 14808 | SIGHUP                      | If the edit buffer has been modified since the last complete write, <i>ex</i> shall attempt to save the edit buffer so that it can be recovered later using the <b>-r</b> option or the <i>ex</i> <b>recover</b> command. The editor shall not write the file or return to command or text input mode, and shall terminate with a non-zero exit status. |
| 14809 |                             |                                                                                                                                                                                                                                                                                                                                                         |
| 14810 |                             |                                                                                                                                                                                                                                                                                                                                                         |
| 14811 |                             |                                                                                                                                                                                                                                                                                                                                                         |
| 14812 | SIGTERM                     | Refer to SIGHUP.                                                                                                                                                                                                                                                                                                                                        |
| 14813 |                             | The action taken for all other signals is unspecified.                                                                                                                                                                                                                                                                                                  |
| 14814 | <b>STDOUT</b>               |                                                                                                                                                                                                                                                                                                                                                         |
| 14815 |                             | The standard output shall be used only for writing prompts to the user, for informational messages, and for writing lines from the file.                                                                                                                                                                                                                |
| 14816 |                             |                                                                                                                                                                                                                                                                                                                                                         |
| 14817 | <b>STDERR</b>               |                                                                                                                                                                                                                                                                                                                                                         |
| 14818 |                             | Used only for diagnostic messages.                                                                                                                                                                                                                                                                                                                      |
| 14819 | <b>OUTPUT FILES</b>         |                                                                                                                                                                                                                                                                                                                                                         |
| 14820 |                             | The output from <i>ex</i> shall be text files.                                                                                                                                                                                                                                                                                                          |
| 14821 | <b>EXTENDED DESCRIPTION</b> |                                                                                                                                                                                                                                                                                                                                                         |
| 14822 |                             | Only the <i>ex</i> mode of the editor is described in this section. See <i>vi</i> for additional editing capabilities available in <i>ex</i> .                                                                                                                                                                                                          |
| 14823 |                             |                                                                                                                                                                                                                                                                                                                                                         |
| 14824 |                             | When an error occurs, <i>ex</i> shall write a message. If the terminal supports a standout mode (such as inverse video), the message shall be written in standout mode. If the terminal does not support a standout mode, and the edit option <b>errorbells</b> is set, an alert action shall precede the error message.                                |
| 14825 |                             |                                                                                                                                                                                                                                                                                                                                                         |
| 14826 |                             |                                                                                                                                                                                                                                                                                                                                                         |
| 14827 |                             |                                                                                                                                                                                                                                                                                                                                                         |
| 14828 |                             | By default, <i>ex</i> shall start in command mode, which shall be indicated by a <b>:</b> prompt; see the <b>prompt</b> command. Text input mode can be entered by the <b>append</b> , <b>insert</b> , or <b>change</b> commands; it can be exited (and command mode re-entered) by typing a period ( <b>.</b> ) alone at the beginning of a line.      |
| 14829 |                             |                                                                                                                                                                                                                                                                                                                                                         |
| 14830 |                             |                                                                                                                                                                                                                                                                                                                                                         |
| 14831 |                             |                                                                                                                                                                                                                                                                                                                                                         |

## Initialization in ex and vi

The following symbols are used in this and following sections to specify locations in the edit buffer:

### *alternate and current path names*

Two path names, named *current* and *alternate*, are maintained by the editor. Any *ex* commands that take file names as arguments shall set them as follows:

1. If a *file* argument is specified to the *ex* **edit**, **ex**, or **recover** commands, or if an *ex* **tag** command replaces the contents of the edit buffer.
  - a. If the command replaces the contents of the edit buffer, the current path name shall be set to the *file* argument or the file indicated by the tag, and the alternate path name shall be set to the previous value of the current path name.
  - b. Otherwise, the alternate path name shall be set to the *file* argument.
2. If a *file* argument is specified to the *ex* **next** command:
  - a. If the command replaces the contents of the edit buffer, the current path name shall be set to the first *file* argument, and the alternate path name shall be set to the previous value of the current path name.
3. If a *file* argument is specified to the *ex* **file** command, the current path name shall be set to the *file* argument, and the alternate path name shall be set to the previous value of the current path name.
4. If a *file* argument is specified to the *ex* **read** and **write** commands (that is, when reading or writing a file, and not to the program named by the **shell** edit option), or a *file* argument is specified to the *ex* **xit** command:
  - a. If the current path name has no value, the current path name shall be set to the *file* argument.
  - b. Otherwise, the alternate path name shall be set to the *file* argument.

If the alternate path name is set to the previous value of the current path name when the current path name had no previous value, then the alternate path name shall have no value as a result.

### *current line*

The line of the edit buffer referenced by the cursor. Each command description specifies the current line after the command has been executed, as the *current line value*. When the edit buffer contains no lines, the current line shall be zero; see **Addressing in ex** on page 394.

### *current column*

The current screen column occupied by the cursor. (The columns shall be numbered beginning at 1.) Each command description specifies the current column after the command has been executed, as the *current column value*. This column is an *ideal* column that is remembered over the lifetime of the editor. The actual screen column upon which the cursor rests may be different from the current column; see the cursor positioning discussion in **Command Descriptions in vi** on page 1031.

### *set to non-<blank>*

A description for a current column value, meaning that the current column shall be set to the last screen column on which is displayed any part of the first non-<blank> character of the line. If the line has no non-<blank> characters, the current column shall be set to the last screen column on which is displayed any part of the last character in the line. If the line is empty, the current column shall be set to column position 1.

14877 The length of lines in the edit buffer may be limited to {LINE\_MAX} bytes. In open and visual  
 14878 mode, the length of lines in the edit buffer may be limited to the number of characters that will  
 14879 fit in the display. If either limit is exceeded during editing, an error message shall be written. If  
 14880 either limit is exceeded by a line read in from a file, an error message shall be written and the  
 14881 edit session may be terminated.

14882 If the editor stops running due to any reason other than a user command, and the edit buffer has  
 14883 been modified since the last complete write, it shall be equivalent to a SIGHUP asynchronous  
 14884 event. If the system crashes, it shall be equivalent to a SIGHUP asynchronous event.

14885 During initialization (before the first file is copied into the edit buffer or any user commands  
 14886 from the terminal are processed) the following shall occur:

- 14887 1. If the environment variable *EXINIT* is set, the editor shall execute the *ex* commands  
 14888 contained in that variable.
- 14889 2. If the *EXINIT* variable is not set, and all of the following are true:
  - 14890 a. The *HOME* environment variable is not null and not empty.
  - 14891 b. The file *.exrc* in the directory referred to by the *HOME* environment variable:
    - 14892 1. Exists
    - 14893 2. Is owned by the same user ID as the real user ID of the process or the process  
 14894 has appropriate privileges
    - 14895 3. Is not writeable by anyone other than the owner

14896 The editor shall execute the *ex* commands contained in that file.

- 14897 3. If and only if all the following are true:
  - 14898 a. The current directory is not referred to by the *HOME* environment variable.
  - 14899 b. A command in the *EXINIT* environment variable or a command in the *.exrc* file in the  
 14900 directory referred to by the *HOME* environment variable sets the editor option *exrc*.
  - 14901 c. The *.exrc* file in the current directory:
    - 14902 1. Exists
    - 14903 2. Is owned by the same user ID as the real user ID of the process, or by one of a  
 14904 set of implementation-dependent user IDs
    - 14905 3. Is not writeable by anyone other than the owner

14906 The editor shall attempt to execute the *ex* commands contained in that file.

14907 Lines in any *.exrc* file that contain no characters or only <blank> characters shall be ignored. If  
 14908 any *.exrc* file exists, but is not read for ownership or permission reasons, it shall be an error.

14909 After the *EXINIT* variable and any *.exrc* files are processed, the first file specified by the user  
 14910 shall be edited, as follows:

- 14911 1. If the user specified the *-t* option, the effect shall be as if the *ex tag* command was entered  
 14912 with the specified argument, with the exception that if tag processing does not result in a  
 14913 file to edit, the effect shall be as described in step 3. below.
- 14914 2. Otherwise, if the user specified any command line *file* arguments, the effect shall be as if  
 14915 the *ex edit* command was entered with the first of those arguments as its *file* argument.
- 14916 3. Otherwise, the effect shall be as if the *ex edit* command was entered with a nonexistent file  
 14917 name as its *file* argument. It is unspecified whether this action shall set the current path

14918 name. In an implementation where this action does not set the current path name, any  
 14919 editor command using the current path name shall fail until an editor command sets the  
 14920 current path name.

14921 If the **-r** option was specified, the first time a file in the initial argument list or a file specified by  
 14922 the **-t** option is edited, if recovery information has previously been saved about it, that  
 14923 information shall be recovered and the editor shall behave as if the contents of the edit buffer  
 14924 have already been modified. If there are multiple instances of the file to be recovered, the one  
 14925 most recently saved shall be recovered, and an informational message that there are previous  
 14926 versions of the file that can be recovered shall be written. If no recovery information about a file  
 14927 is available, an informational message to this effect shall be written, and the edit shall proceed as  
 14928 usual.

14929 If the **-** option was specified, the first time a file that already exists (including a file that might  
 14930 not exist but for which recovery information is available, when the **-r** option is specified)  
 14931 replaces or initializes the contents of the edit buffer, the current line shall be set to the last line of  
 14932 the edit buffer, the current column shall be set to non-**<blank>**, and the **ex** commands specified  
 14933 with the **-c** option shall be executed. In this case, the current line and current column shall not be  
 14934 set as described for the command associated with the replacement or initialization of the edit  
 14935 buffer contents. However, if the **-t** option or a **tag** command is associated with this action, the **-c**  
 14936 option commands shall be executed and then the movement to the tag shall be performed.

14937 The current argument list shall initially be set to the file names specified by the user on the  
 14938 command line. If no file names are specified by the user, the current argument list shall be  
 14939 empty. If the **-t** option was specified, it is unspecified whether any file name resulting from tag  
 14940 processing shall be prepended to the current argument list. In the case where the file name is  
 14941 added as a prefix to the current argument list, the current argument list reference shall be set to  
 14942 that file name. In the case where the file name is not added as a prefix to the current argument  
 14943 list, the current argument list reference shall logically be located before the first of the file names  
 14944 specified on the command line (for example, a subsequent **ex next** command shall edit the first  
 14945 file name from the command line). If the **-t** option was not specified, the current argument list  
 14946 reference shall be to the first of the file names on the command line.

#### 14947 Addressing in ex

14948 Addressing in **ex** relates to the current line and the current column; the address of a line is its 1-  
 14949 based line number, the address of a column is its 1-based count from the beginning of the line.  
 14950 Generally, the current line is the last line affected by a command. The current line number is the  
 14951 address of the current line. In each command description, the effect of the command on the  
 14952 current line number and the current column is described.

14953 Addresses are constructed as follows:

- 14954 1. The character **'.'** (period) shall address the current line.
- 14955 2. The character **'\$'** shall address the last line of the edit buffer.
- 14956 3. The positive decimal number *n* shall address the *n*th line of the edit buffer.
- 14957 4. The address **"'x"** refers to the line marked with the mark name character **'x'**, which shall  
 14958 be a lowercase letter from the portable character set or one of the characters **'\''** or **'\''**. It  
 14959 shall be an error if the line that was marked is not currently present in the edit buffer or the  
 14960 mark has not been set. Lines can be marked with the **ex mark** or **k** commands, or the **vi m**  
 14961 command.
- 14962 5. A regular expression (RE) enclosed by slashes (**'/'**) shall address the first line found by  
 14963 searching forwards from the line following the current line toward the end of the edit

- 14964 buffer and stopping at the first line containing a string matching the regular expression. As  
 14965 stated in **Regular Expressions in ex** on page 424, an address consisting of a null regular  
 14966 expression delimited by slashes "/" shall address the next line containing the last regular  
 14967 expression encountered. In addition, the second slash can be omitted at the end of a  
 14968 command line. If the **wrapsan** edit option is set, the search shall wrap around to the  
 14969 beginning of the edit buffer and continue up to and including the current line, so that the  
 14970 entire edit buffer is searched. Within the regular expression, the sequence "\/" shall  
 14971 represent a literal slash instead of the regular expression delimiter.
- 14972 6. A regular expression enclosed in question marks ('?') shall address the first line found by  
 14973 searching backwards from the line preceding the current line toward the beginning of the  
 14974 edit buffer and stopping at the first line containing a string matching the regular  
 14975 expression. The second question mark can be omitted at the end of a command line. If the  
 14976 **wrapsan** edit option is set, the search shall wrap around from the beginning of the edit  
 14977 buffer to the end of the edit buffer and continue up to and including the current line, so  
 14978 that the edit entire buffer is searched. Within the regular expression, the sequence "\?"  
 14979 shall represent a literal question mark instead of the RE delimiter.
- 14980 7. A plus sign ('+') or a minus sign ('-') followed by a decimal number shall address the  
 14981 current line plus or minus the number. A '+' or '-' not followed by a decimal number  
 14982 shall address the current line plus or minus 1.
- 14983 Addresses can be followed by zero or more address offsets, optionally <blank> character-  
 14984 separated. Address offsets are constructed as follows:
- 14985 1. A '+' or '-' immediately followed by a decimal number shall add (subtract) the  
 14986 indicated number of lines to (from) the address. A '+' or '-' not followed by a decimal  
 14987 number shall add (subtract) 1 to (from) the address.
- 14988 2. A decimal number shall add the indicated number of lines to the address.
- 14989 It shall not be an error for an intermediate address value to be less than zero or greater than the  
 14990 last line in the edit buffer. It shall be an error for the final address value to be less than zero or  
 14991 greater than the last line in the edit buffer.
- 14992 Commands take zero, one, or two addresses; see the descriptions of *1addr* and *2addr* in  
 14993 **Command Descriptions in ex** on page 401. If more than the required number of addresses are  
 14994 provided to a command that requires zero addresses, it shall be an error. Otherwise, if more than  
 14995 the required number of addresses are provided to a command, the addresses specified first shall  
 14996 be evaluated and then discarded until the maximum number of valid addresses remain.
- 14997 Addresses shall be separated from each other by a comma (',') or a semicolon (';'). If no  
 14998 address is specified before or after a comma or semicolon separator, it shall be as if the address  
 14999 of the current line was specified before or after the separator. In the case of a semicolon  
 15000 separator, the current line ('.') shall be set to the first address, and only then will the next  
 15001 address be calculated. This feature can be used to determine the starting line for forwards and  
 15002 backwards searches (see rules 5. and 6.).
- 15003 A percent sign ('%') shall be equivalent to entering the two addresses "1,\$".
- 15004 Any delimiting <blank> characters between addresses, address separators, or address offsets  
 15005 shall be discarded.

## Command Line Parsing in ex

The following symbol is used in this and following sections to describe parsing behavior:

*escape* If a character is referred to as “backslash escaped” or “<control>-V escaped,” it shall mean that the character acquired or lost a special meaning by virtue of being preceded, respectively, by a backslash or <control>-V character. Unless otherwise specified, the escaping character shall be discarded at that time and shall not be further considered for any purpose.

Command-line parsing shall be done in the following steps. For each step, characters already evaluated shall be ignored; that is, the phrase “leading character” refers to the next character that has not yet been evaluated.

1. Leading colon characters shall be skipped.
2. Leading <blank> characters shall be skipped.
3. If the leading character is a double-quote character, the characters up to and including the next non-backslash-escaped <newline> character shall be discarded, and any subsequent characters shall be parsed as a separate command.
4. Leading characters that can be interpreted as addresses shall be evaluated; see **Addressing in ex** on page 394.
5. Leading <blank> characters shall be skipped.
6. If the next character is a vertical-line character or a <newline> character:
  - a. If the next character is a <newline> character:
    1. If *ex* is in open or visual mode, the current line shall be set to the last address specified, if any.
    2. Otherwise, if the last command was terminated by a vertical-line character, no action shall be taken; for example, the command "||<newline>" shall execute two implied commands, not three.
    3. Otherwise, step 6.b. shall apply.
  - b. Otherwise, the implied command shall be the **print** command. The last #, **p**, and **l** flags specified to any *ex* command shall be remembered and shall apply to this implied command. Executing the *ex* **number**, **print**, or **list** command shall set the remembered flags to #, nothing, and **l**, respectively, plus any other flags specified for that execution of the **number**, **print**, or **list** command.
 

If *ex* is not currently performing a **global** or **v** command, and no address or count is specified, the current line shall be incremented by 1 before the command is executed. If incrementing the current line would result in an address past the last line in the edit buffer, the command shall fail, and the increment shall not happen.
  - c. The <newline> character or vertical-line character shall be discarded and any subsequent characters shall be parsed as a separate command.
7. The command name shall be comprised of the next character (if the character is not alphabetic), or the next character and any subsequent alphabetic characters (if the character is alphabetic), with the following exceptions:
  - a. Commands that consist of any prefix of the characters in the command name **delete**, followed immediately by any of the characters **l**, **p**, **+**, **-**, or **#** shall be interpreted as a **delete** command, followed by a <blank> character, followed by the characters that



were not part of the prefix of the **delete** command. The maximum number of characters shall be matched to the command name **delete**; for example, "de1" shall not be treated as "de" followed by the flag **l**.

b. Commands that consist of the character **k**, followed by a character that can be used as the name of a mark, shall be equivalent to the mark command followed by a <blank> character, followed by the character that followed the **k**.

c. Commands that consist of the character **s**, followed by characters that could be interpreted as valid options to the **s** command, shall be the equivalent of the **s** command, without any pattern or replacement values, followed by a <blank> character, followed by the characters after the **s**.

8. The command name shall be matched against the possible command names, and a command name that contains a prefix matching the characters specified by the user shall be the executed command. In the case of commands where the characters specified by the user could be ambiguous, the executed command shall be as follows:

|           |               |           |              |           |              |
|-----------|---------------|-----------|--------------|-----------|--------------|
| <b>a</b>  | <b>append</b> | <b>n</b>  | <b>next</b>  | <b>t</b>  | <b>t</b>     |
| <b>c</b>  | <b>change</b> | <b>p</b>  | <b>print</b> | <b>u</b>  | <b>undo</b>  |
| <b>ch</b> | <b>change</b> | <b>pr</b> | <b>print</b> | <b>un</b> | <b>undo</b>  |
| <b>e</b>  | <b>edit</b>   | <b>r</b>  | <b>read</b>  | <b>v</b>  | <b>v</b>     |
| <b>m</b>  | <b>move</b>   | <b>re</b> | <b>read</b>  | <b>w</b>  | <b>write</b> |
| <b>ma</b> | <b>mark</b>   | <b>s</b>  | <b>s</b>     |           |              |

Implementation extensions with names causing similar ambiguities shall not be checked for a match until all possible matches for commands specified by IEEE Std. 1003.1-200x have been checked.

9. If the command is a **!** command, or if the command is a **read** command followed by zero or more <blank> characters and a **!**, or if the command is a **write** command followed by one or more <blank> characters and a **!**, the rest of the command shall include all characters up to a non-backslash-escaped <newline> character. The <newline> character shall be discarded and any subsequent characters shall be parsed as a separate *ex* command.

10. Otherwise, if the command is an **edit**, **ex**, or **next** command, or a **visual** command while in open or visual mode, the next part of the command shall be parsed as follows:

a. Any '!' character immediately following the command shall be skipped and be part of the command.

b. Any leading <blank> characters shall be skipped and be part of the command.

c. If the next character is a '+', characters up to the first non-backslash-escaped <newline> character or non-backslash-escaped <blank> character shall be skipped and be part of the command.

d. The rest of the command shall be determined by the steps specified in paragraph 12.

11. Otherwise, if the command is a **global**, **open**, **s**, or **v** command, the next part of the command shall be parsed as follows:

a. Any leading <blank> characters shall be skipped and be part of the command.

b. If the next character is not an alphanumeric, double-quote, <newline>, backslash, or vertical-line character:

1. The next character shall be used as a command delimiter.

- 15093                   2. If the command is a **global**, **open**, or **v** command, characters up to the first  
15094 non-backslash-escaped <newline> character, or first non-backslash-escaped  
15095 delimiter character, shall be skipped and be part of the command.
- 15096                   3. If the command is an **s** command, characters up to the first non-backslash-  
15097 escaped <newline> character, or second non-backslash-escaped delimiter  
15098 character, shall be skipped and be part of the command.
- 15099           c. If the command is a **global** or **v** command, characters up to the first non-backslash-  
15100 escaped <newline> character shall be skipped and be part of the command.
- 15101           d. Otherwise, the rest of the command shall be determined by the steps specified in  
15102 paragraph 12.

15103       12. Otherwise:

- 15104           a. If the command was a **map**, **unmap**, **abbreviate**, or **unabbreviate** command,  
15105 characters up to the first non-<control>-V-escaped <newline>, vertical-line, or  
15106 double-quote character shall be skipped and be part of the command.
- 15107           b. Otherwise, characters up to the first non-backslash-escaped <newline>, vertical-line,  
15108 or double-quote character shall be skipped and be part of the command.
- 15109           c. If the command was an **append**, **change**, or **insert** command, and the step 12.b.  
15110 ended at a vertical-line character, any subsequent characters, up to the next non-  
15111 backslash-escaped <newline> character shall be used as input text to the command.
- 15112           d. If the command was ended by a double-quote character, all subsequent characters,  
15113 up to the next non-backslash-escaped <newline> character, shall be discarded.
- 15114           e. The terminating <newline> or vertical-line character shall be discarded and any  
15115 subsequent characters shall be parsed as a separate **ex** command.

15116       Command arguments shall be parsed as described by the Synopsis and Description of each  
15117 individual **ex** command. This parsing shall not be <blank> character-sensitive, except for the **!**  
15118 argument, which must follow the command name without intervening <blank> characters, and  
15119 where it would otherwise be ambiguous. For example, *count* and *flag* arguments need not be  
15120 <blank> character separated because "d22p" is not ambiguous, but *file* arguments to the **ex next**  
15121 command must be separated by one or more <blank> characters. Any <blank> character in  
15122 command arguments for the **abbreviate**, **unabbreviate**, **map**, and **unmap** commands can be  
15123 <control>-V-escaped, in which case the <blank> character shall not be used as an argument  
15124 delimiter. Any <blank> character in the command argument for any other command can be  
15125 backslash-escaped, in which case that <blank> character shall not be used as an argument  
15126 delimiter.

15127       Within command arguments for the **abbreviate**, **unabbreviate**, **map**, and **unmap** commands,  
15128 any character can be <control>-V-escaped. All such escaped characters shall be treated literally  
15129 and shall have no special meaning. Within command arguments for all other **ex** commands that  
15130 are not regular expressions or replacement strings, any character that would otherwise have a  
15131 special meaning can be backslash-escaped. Escaped characters shall be treated literally, without  
15132 special meaning as shell expansion characters or '**!**', '**%**', and '**#**' expansion characters. See  
15133 **Regular Expressions in ex** on page 424 and **Replacement Strings in ex** on page 425 for  
15134 descriptions of command arguments that are regular expressions or replacement strings.

15135       Non-backslash-escaped '**%**' characters appearing in *file* arguments to any **ex** command shall be  
15136 replaced by the current path name; unescaped '**#**' characters shall be replaced by the alternate  
15137 path name. It shall be an error if '**%**' or '**#**' characters appear unescaped in an argument and  
15138 their corresponding values are not set.

Non-backslash-escaped '!' characters in the arguments to either the **ex!** command or the open and visual mode **!** command, or in the arguments to the **ex read** command, where the first non-**<blank>** character after the command name is a '!' character, or in the arguments to the **ex write** command where the command name is followed by one or more **<blank>** characters and the first non-**<blank>** character after the command name is a '!' character, shall be replaced with the arguments to the last of those three commands as they appeared after all unescaped '%', '#', and '!' characters were replaced. It shall be an error if '!' characters appear unescaped in one of these commands and there has been no previous execution of one of these commands.

If an error occurs during the parsing or execution of an **ex** command:

- An informational message to this effect shall be written. Execution of the **ex** command shall stop, and the cursor (for example, the current line and column) shall not be further modified.
- If the **ex** command resulted from a map expansion, all characters from that map expansion shall be discarded, except as otherwise specified by the **map** command.
- Otherwise, if the **ex** command resulted from the processing of an **EXINIT** environment variable, a **.exrc** file, a **:source** command, a **-c** option, or a **+command** specified to an **ex edit**, **ex**, **next**, or **visual** command, no further commands from the source of the commands shall be executed.
- Otherwise, if the **ex** command resulted from the execution of a buffer or a **global** or **v** command, no further commands caused by the execution of the buffer or the **global** or **v** command shall be executed.
- Otherwise, if the **ex** command was not terminated by a **<newline>** character, all characters up to and including the next non-backslash-escaped **<newline>** character shall be discarded.

### Input Editing in **ex**

The following symbols are used in this and following sections to specify command actions.

**word** In the POSIX locale, a word consists of a maximal sequence of letters, digits, and underscores, delimited at both ends by characters other than letters, digits, or underscores, or by the beginning or end of a line or the edit buffer.

When accepting input characters from the user, in either **ex** command mode or **ex** text input mode, **ex** shall enable canonical mode input processing, as defined in the System Interfaces volume of IEEE Std. 1003.1-200x.

If in **ex** text input mode:

1. If the **number** edit option is set, **ex** shall prompt for input using the line number that would be assigned to the line if it is entered, in the format specified for the **ex number** command.
2. If the **autoindent** edit option is set, **ex** shall prompt for input using **autoindent** characters, as described by the **autoindent** edit option. **autoindent** characters shall follow the line number, if any.

If in **ex** command mode:

1. If the **prompt** edit option is set, input shall be prompted for using a single ':' character; otherwise, there shall be no prompt.

The input characters in the following sections shall have the following effects on the input line.

|       |                                                                                                              |  |
|-------|--------------------------------------------------------------------------------------------------------------|--|
| 15180 | <b>Scroll</b>                                                                                                |  |
| 15181 | <i>Synopsis:</i> <code>eof</code>                                                                            |  |
| 15182 | See the description of the <i>stty eof</i> character in <i>stty</i> .                                        |  |
| 15183 | If in <i>ex</i> command mode:                                                                                |  |
| 15184 | If the <i>eof</i> character is the first character entered on the line, the line shall be evaluated as if it |  |
| 15185 | contained two characters: a <control>-D and a <newline> character.                                           |  |
| 15186 | Otherwise, the <i>eof</i> character shall have no special meaning.                                           |  |
| 15187 | If in <i>ex</i> text input mode:                                                                             |  |
| 15188 | If the cursor follows an <b>autoindent</b> character, the <b>autoindent</b> characters in the line shall be  |  |
| 15189 | modified so that a part of the next text input character will be displayed on the first column               |  |
| 15190 | in the line after the previous <b>shiftwidth</b> edit option column boundary, and the user shall be          |  |
| 15191 | prompted again for input for the same line.                                                                  |  |
| 15192 | Otherwise, if the cursor follows a '0', which follows an <b>autoindent</b> character, and the '0'            |  |
| 15193 | was the previous text input character, the '0' and all <b>autoindent</b> characters in the line shall        |  |
| 15194 | be discarded, and the user shall be prompted again for input for the same line.                              |  |
| 15195 | Otherwise, if the cursor follows a '^', which follows an <b>autoindent</b> character, and the '^'            |  |
| 15196 | was the previous text input character, the '^' and all <b>autoindent</b> characters in the line shall        |  |
| 15197 | be discarded, and the user shall be prompted again for input for the same line. In addition,                 |  |
| 15198 | the <b>autoindent</b> level for the next input line shall be derived from the same line from which           |  |
| 15199 | the <b>autoindent</b> level for the current input line was derived.                                          |  |
| 15200 | Otherwise, if there are no <b>autoindent</b> or text input characters in the line, the <i>eof</i> character  |  |
| 15201 | shall be discarded.                                                                                          |  |
| 15202 | Otherwise, the <i>eof</i> character shall have no special meaning.                                           |  |
| 15203 | <b>&lt;newline&gt;</b>                                                                                       |  |
| 15204 | <i>Synopsis:</i> <newline>                                                                                   |  |
| 15205 | <control>-J                                                                                                  |  |
| 15206 | If in <i>ex</i> command mode:                                                                                |  |
| 15207 | Cause the command line to be parsed; <control>-J shall be mapped to the <newline>                            |  |
| 15208 | character for this purpose.                                                                                  |  |
| 15209 | If in <i>ex</i> text input mode:                                                                             |  |
| 15210 | Terminate the current line. If there are no characters other than <b>autoindent</b> characters on the        |  |
| 15211 | line, all characters on the line shall be discarded.                                                         |  |
| 15212 | Prompt for text input on a new line after the current line. If the <b>autoindent</b> edit option is set,     |  |
| 15213 | an appropriate number of <b>autoindent</b> characters shall be added as a prefix to the line as              |  |
| 15214 | described by the <i>ex</i> <b>autoindent</b> edit option.                                                    |  |

|       |                                                                                                  |                                                                                                                                        |
|-------|--------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| 15215 | <b>&lt;backslash&gt;</b>                                                                         |                                                                                                                                        |
| 15216 | <i>Synopsis:</i>                                                                                 | <backslash>                                                                                                                            |
| 15217 | Allow the entry of a subsequent <newline> or <control>-J as a literal character, removing any    |                                                                                                                                        |
| 15218 | special meaning that it may have to the editor during text input mode. The backslash character   |                                                                                                                                        |
| 15219 | shall be retained and evaluated when the command line is parsed, or retained and included        |                                                                                                                                        |
| 15220 | when the input text becomes part of the edit buffer.                                             |                                                                                                                                        |
| 15221 | <b>&lt;control&gt;-V</b>                                                                         |                                                                                                                                        |
| 15222 | <i>Synopsis:</i>                                                                                 | <control>-V                                                                                                                            |
| 15223 | Allow the entry of any subsequent character as a literal character, removing any special meaning |                                                                                                                                        |
| 15224 | that it may have to the editor during text input mode. The <control>-V character shall be        |                                                                                                                                        |
| 15225 | discarded before the command line is parsed or the input text becomes part of the edit buffer.   |                                                                                                                                        |
| 15226 | If the “literal next” functionality is performed by the underlying system, it is implementation- |                                                                                                                                        |
| 15227 | dependent whether a character other than <control>-V performs this function.                     |                                                                                                                                        |
| 15228 | <b>&lt;control&gt;-W</b>                                                                         |                                                                                                                                        |
| 15229 | <i>Synopsis:</i>                                                                                 | <control>-W                                                                                                                            |
| 15230 | Discard the <control>-W, and the word previous to it in the input line, including any <blank>    |                                                                                                                                        |
| 15231 | characters following the word and preceding the <control>-W. If the “word erase” functionality   |                                                                                                                                        |
| 15232 | is performed by the underlying system, it is implementation-dependent whether a character        |                                                                                                                                        |
| 15233 | other than <control>-W performs this function.                                                   |                                                                                                                                        |
| 15234 | <b>Command Descriptions in ex</b>                                                                |                                                                                                                                        |
| 15235 | The following symbols are used in this section to represent command modifiers. Some of these     |                                                                                                                                        |
| 15236 | modifiers can be omitted, in which case the specified defaults shall be used.                    |                                                                                                                                        |
| 15237 | <i>1addr</i>                                                                                     | A single line address, given in any of the forms described in <b>Addressing in ex</b> on                                               |
| 15238 |                                                                                                  | page 394; the default shall be the current line ( ' . ' ), unless otherwise specified.                                                 |
| 15239 |                                                                                                  | If the line address is zero, it shall be an error, unless otherwise specified in the                                                   |
| 15240 |                                                                                                  | following command descriptions.                                                                                                        |
| 15241 |                                                                                                  | If the edit buffer is empty, and the address is specified with a command other than                                                    |
| 15242 |                                                                                                  | =, <b>append</b> , <b>insert</b> , <b>open</b> , <b>put</b> , <b>read</b> , or <b>visual</b> , or the address is not zero, it shall be |
| 15243 |                                                                                                  | an error.                                                                                                                              |
| 15244 | <i>2addr</i>                                                                                     | Two addresses specifying an inclusive range of lines. If no addresses are specified,                                                   |
| 15245 |                                                                                                  | the default for <i>2addr</i> shall be the current line only ( " . , . " ), unless otherwise                                            |
| 15246 |                                                                                                  | specified in the following command descriptions. If one address is specified, <i>2addr</i>                                             |
| 15247 |                                                                                                  | shall specify that line only, unless otherwise specified in the following command                                                      |
| 15248 |                                                                                                  | descriptions.                                                                                                                          |
| 15249 |                                                                                                  | It shall be an error if the first address is greater than the second address.                                                          |
| 15250 |                                                                                                  | If the edit buffer is empty, and the two addresses are specified with a command                                                        |
| 15251 |                                                                                                  | other than the <b>!</b> , <b>write</b> , <b>wq</b> , or <b>xit</b> commands, or either address is not zero, it shall                   |
| 15252 |                                                                                                  | be an error.                                                                                                                           |
| 15253 | <i>count</i>                                                                                     | A positive decimal number. If <i>count</i> is specified, it shall be equivalent to specifying                                          |
| 15254 |                                                                                                  | an additional address to the command, unless otherwise specified by the following                                                      |
| 15255 |                                                                                                  | command descriptions. The additional address shall be equal to the last address                                                        |
| 15256 |                                                                                                  | specified to the command (either explicitly or by default) plus <i>count</i> −1.                                                       |

|       |               |                                                                                                                                  |
|-------|---------------|----------------------------------------------------------------------------------------------------------------------------------|
| 15257 |               | If this would result in an address greater than the last line of the edit buffer, it shall                                       |
| 15258 |               | be corrected to equal the last line of the edit buffer.                                                                          |
| 15259 | <i>flags</i>  | One or more of the characters '+' , '-' , '#' , 'p' , or 'l' (ell). The flag characters                                          |
| 15260 |               | can be <blank>-separated, and in any order or combination. The characters '#' ,                                                  |
| 15261 |               | 'p' , and 'l' shall cause lines to be written in the format specified by the <b>print</b>                                        |
| 15262 |               | command with the specified <i>flags</i> .                                                                                        |
| 15263 |               | The lines to be written are as follows:                                                                                          |
| 15264 |               | 1. All edit buffer lines written during the execution of the <i>ex</i> &, ~, <b>list</b> , <b>number</b> ,                       |
| 15265 |               | <b>open</b> , <b>print</b> , <b>s</b> , <b>visual</b> , and <b>z</b> commands shall be written as specified by <i>flags</i> .    |
| 15266 |               | 2. After the completion of an <i>ex</i> command with a flag as an argument, the                                                  |
| 15267 |               | current line shall be written as specified by <i>flags</i> , unless the current line was                                         |
| 15268 |               | the last line written by the command.                                                                                            |
| 15269 |               | The characters '+' and '-' cause the value of the current line after the execution                                               |
| 15270 |               | of the <i>ex</i> command to be adjusted by the offset address as described in <b>Addressing</b>                                  |
| 15271 |               | <b>in ex</b> on page 394. This adjustment shall occur before the current line is written as                                      |
| 15272 |               | described in 2. above.                                                                                                           |
| 15273 |               | The default for <i>flags</i> shall be none.                                                                                      |
| 15274 | <i>buffer</i> | One of a number of named areas for holding text. The named buffers are specified                                                 |
| 15275 |               | by the alphanumeric characters of the POSIX locale. There shall also be one                                                      |
| 15276 |               | "unnamed" buffer. When no buffer is specified for editor commands that use a                                                     |
| 15277 |               | buffer, the unnamed buffer shall be used. Commands that store text into buffers                                                  |
| 15278 |               | shall store the text as it was before the command took effect, and shall store text                                              |
| 15279 |               | occurring earlier in the file before text occurring later in the file, regardless of how                                         |
| 15280 |               | the text region was specified. Commands that store text into buffers shall store the                                             |
| 15281 |               | text into the unnamed buffer as well as any specified buffer.                                                                    |
| 15282 |               | In <i>ex</i> commands, buffer names are specified as the name by itself. In open or visual                                       |
| 15283 |               | mode commands the name is preceded by a double quote ( ' ' ) character.                                                          |
| 15284 |               | If the specified buffer name is an uppercase character, and the buffer contents are                                              |
| 15285 |               | to be modified, the buffer shall be appended to rather than being overwritten. If                                                |
| 15286 |               | the buffer is not being modified, specifying the buffer name in lowercase and                                                    |
| 15287 |               | uppercase shall have identical results.                                                                                          |
| 15288 |               | There shall also be buffers named by the numbers 1 through 9. In open and visual                                                 |
| 15289 |               | mode, if a region of text including characters from more than a single line is being                                             |
| 15290 |               | modified by the <i>vi</i> <b>c</b> or <b>d</b> commands, the motion character associated with the <b>c</b> or                    |
| 15291 |               | <b>d</b> commands specifies that the buffer text shall be in line mode, or the commands                                          |
| 15292 |               | %, ' , / , ? , ( , ) , <b>N</b> , <b>n</b> , { , or } are used to define a region of text for the <b>c</b> or <b>d</b> commands, |
| 15293 |               | the contents of buffers 1 through 8 shall be moved into the buffer named by the                                                  |
| 15294 |               | next numerically greater value, the contents of buffer 9 shall be discarded, and the                                             |
| 15295 |               | region of text shall be copied into buffer 1. This shall be in addition to copying the                                           |
| 15296 |               | text into a user-specified buffer or unnamed buffer, or both. Numeric buffers can                                                |
| 15297 |               | be specified as a source buffer for open and visual mode commands; however,                                                      |
| 15298 |               | specifying a numeric buffer as the write target of an open or visual mode                                                        |
| 15299 |               | command shall have unspecified results.                                                                                          |
| 15300 |               | The text of each buffer shall have the characteristic of being in either line or                                                 |
| 15301 |               | character mode. Appending text to a non-empty buffer shall set the mode to match                                                 |
| 15302 |               | the characteristic of the text being appended. Appending text to a buffer shall                                                  |
| 15303 |               | cause the creation of at least one additional line in the buffer. All text stored into                                           |

15304 buffers by *ex* commands shall be in line mode. The *ex* commands that use buffers  
 15305 as the source of text specify individually how buffers of different modes are  
 15306 handled. Each open or visual mode command that uses buffers for any purpose  
 15307 specifies individually the mode of the text stored into the buffer and how buffers  
 15308 of different modes are handled.

15309 *file* Command text used to derive a path name. The default shall be the current path  
 15310 name, as defined previously, in which case, if no current path name has yet been  
 15311 established it shall be an error, except where specifically noted in the individual  
 15312 command descriptions that follow. If the command text contains any of the  
 15313 characters '~', '{', '[', '\*', '?', '\$', '\', ' ', , and '\', it shall be  
 15314 subjected to the process of "shell expansions", as described below; if more than a  
 15315 single path name results and the command expects only one, it shall be an error.

15316 The process of shell expansions in the editor shall be done as follows. The *ex* utility  
 15317 shall pass two arguments to the program named by the shell edit option; the first  
 15318 shall be *-c*, and the second shall be the string "echo" and the command text as a  
 15319 single argument. The standard output and standard error of that command shall  
 15320 replace the command text.

15321 *!* A character that can be appended to the command name to modify its operation,  
 15322 as detailed in the individual command descriptions. With the exception of the *ex*  
 15323 *read*, *write*, and *!* commands, the '!' character shall only act as a modifier if there  
 15324 are no <blank> characters between it and the command name.

15325 *remembered search direction*

15326 The *vi* commands *N* and *n* begin searching in a forwards or backwards direction in  
 15327 the edit buffer based on a remembered search direction, which is initially unset,  
 15328 and is set by the *ex* *global*, *v*, *s*, and *tag* commands, and the *vi* / and ? commands.

## 15329 Abbreviate

15330 *Synopsis:* *ab[breviate][lhs rhs]*

15331 If *lhs* and *rhs* are not specified, write the current list of abbreviations and do nothing more.

15332 Implementations may restrict the set of characters accepted in *lhs* or *rh*, except that printable  
 15333 characters and <blank> characters shall not be restricted. Additional restrictions shall be  
 15334 implementation-dependent.

15335 In both *lhs* and *rhs*, any character may be escaped with a <control>-V, in which case the  
 15336 character shall not be used to delimit *lhs* from *rhs*, and the escaping <control>-V shall be  
 15337 discarded.

15338 In open and visual text input mode, if a non-word or <ESC> character that is not escaped by a  
 15339 <control>-V character is entered after a word character, a check shall be made for a set of  
 15340 characters matching *lhs*, in the text input entered during this command. If it is found, the effect  
 15341 shall be as if *rhs* was entered instead of *lhs*.

15342 The set of characters that are checked is defined as follows:

- 15343 1. If there are no characters inserted before the word and non-word or <ESC> characters that  
 15344 triggered the check, the set of characters shall consist of the word character.
- 15345 2. If the character inserted before the word and non-word or <ESC> characters that triggered  
 15346 the check is a word character, the set of characters shall consist of the characters inserted  
 15347 immediately before the triggering characters that are word characters, plus the triggering  
 15348 word character.

3. If the character inserted before the word and non-word or <ESC> characters that triggered the check is not a word character, the set of characters shall consist of the characters that were inserted before the triggering characters that are neither <blank> characters nor word characters, plus the triggering word character.

It is unspecified whether the *lhs* argument entered for the **ex** **abbreviate** and **unabbreviate** commands is replaced in this fashion. Regardless of whether or not the replacement occurs, the effect of the command shall be as if the replacement had not occurred.

*Current line:* Unchanged.

*Current column:* Unchanged.

## Append

*Synopsis:*     [*laddr*] a[ppend][!]

Enter text input mode; the input text shall be placed after the specified line. If line zero is specified, the text shall be placed at the beginning of the edit buffer.

This command shall be affected by the **number** and **autoindent** edit options; following the command name with '!' shall cause the **autoindent** edit option setting to be toggled for the duration of this command only.

*Current line:* Set to the last input line; if no lines were input, set to the specified line, or to the first line of the edit buffer if a line of zero was specified, or zero if the edit buffer is empty.

*Current column:* Set to non-<blank>.

## Arguments

*Synopsis:*     ar[gs]

Write the current argument list, with the current argument-list entry, if any, between '[' and ']' characters.

*Current line:* Unchanged.

*Current column:* Unchanged.

## Change

*Synopsis:*     [*2addr*] c[hange][!][count]

Enter **ex** text input mode; the input text shall replace the specified lines. The specified lines shall be copied into the unnamed buffer, which shall become a line mode buffer.

This command shall be affected by the **number** and **autoindent** edit options; following the command name with '!' shall cause the **autoindent** edit option setting to be toggled for the duration of this command only.

*Current line:* Set to the last input line; if no lines were input, set to the line before the first address, or to the first line of the edit buffer if there are no lines preceding the first address, or to zero if the edit buffer is empty.

*Current column:* Set to non-<blank>.



15385 **Change Directory**

15386 *Synopsis:*     chd[ir][!][*directory*]  
 15387               cd[!][*directory*]

15388     Change the current working directory to *directory*.

15389     If no *directory* argument is specified, and the *HOME* environment variable is set to a non-null  
 15390     and non-empty value, *directory* shall default to the value named in the *HOME* environment  
 15391     variable. If the *HOME* environment variable is empty or is undefined, the default value of  
 15392     *directory* is implementation-dependent.

15393     If no '!' is appended to the command name, and the edit buffer has been modified since the  
 15394     last complete write, and the current path name does not begin with a '/', it shall be an error.

15395     *Current line:* Unchanged.

15396     *Current column:* Unchanged.

15397 **Copy**

15398 *Synopsis:*     [*2addr*] co[py] *laddr* [*flags*]  
 15399               [*2addr*] t *laddr* [*flags*]

15400     Copy the specified lines after the specified destination line; line zero specifies that the lines shall  
 15401     be placed at the beginning of the edit buffer.

15402     *Current line:* Set to the last line copied.

15403     *Current column:* Set to non-<blank>.

15404 **Delete**

15405 *Synopsis:*     [*2addr*] d[ele]te[!][*buffer*][*count*][*flags*]

15406     Delete the specified lines into a buffer (defaulting to the unnamed buffer), which shall become a  
 15407     line-mode buffer.

15408     Flags can immediately follow the command name; see **Command Line Parsing in ex** on page  
 15409     396.

15410     *Current line:* Set to the line following the deleted lines, or to the last line in the edit buffer if that  
 15411     line is past the end of the edit buffer, or to zero if the edit buffer is empty.

15412     *Current column:* Set to non-<blank>.

15413 **Edit**

15414 *Synopsis:*     e[dit][!][+*command*][*file*]  
 15415               ex[!][+*command*][*file*]

15416     If no '!' is appended to the command name, and the edit buffer has been modified since the  
 15417     last complete write, it shall be an error.

15418     If *file* is specified, replace the current contents of the edit buffer with the current contents of *file*,  
 15419     and set the current path name to *file*. If *file* is not specified, replace the current contents of the  
 15420     edit buffer with the current contents of the file named by the current path name. If for any  
 15421     reason the current contents of the file cannot be accessed, the edit buffer shall be empty.

15422     The +*command* option shall be <blank> character-delimited; <blank> characters within  
 15423     +*command* can be escaped by preceding them with a backslash character. The +*command* shall be  
 15424     interpreted as an *ex* command immediately after the contents of the edit buffer have been

|       |                                                                                                                |  |
|-------|----------------------------------------------------------------------------------------------------------------|--|
| 15425 | replaced and the current line and column have been set.                                                        |  |
| 15426 | If the edit buffer is empty:                                                                                   |  |
| 15427 | <i>Current line</i> : Set to 0.                                                                                |  |
| 15428 | <i>Current column</i> : Set to 1.                                                                              |  |
| 15429 | Otherwise, if executed while in <i>ex</i> command mode or if the <i>+command</i> argument is specified:        |  |
| 15430 | <i>Current line</i> : Set to the last line of the edit buffer.                                                 |  |
| 15431 | <i>Current column</i> : Set to non-<blank>.                                                                    |  |
| 15432 | Otherwise, if <i>file</i> is omitted or results in the current path name:                                      |  |
| 15433 | <i>Current line</i> : Set to the first line of the edit buffer.                                                |  |
| 15434 | <i>Current column</i> : Set to non-<blank>.                                                                    |  |
| 15435 | Otherwise, if <i>file</i> is the same as the last file edited, the line and column shall be set as follows; if |  |
| 15436 | the file was previously edited, the line and column may be set as follows:                                     |  |
| 15437 | <i>Current line</i> : Set to the last value held when that file was last edited. If this value is not a valid  |  |
| 15438 | line in the new edit buffer, set to the first line of the edit buffer.                                         |  |
| 15439 | <i>Current column</i> : If the current line was set to the last value held when the file was last edited, set  |  |
| 15440 | to the last value held when the file was last edited. Otherwise, or if the last value is not a valid           |  |
| 15441 | column in the new edit buffer, set to non-<blank>.                                                             |  |
| 15442 | Otherwise:                                                                                                     |  |
| 15443 | <i>Current line</i> : Set to the first line of the edit buffer.                                                |  |
| 15444 | <i>Current column</i> : Set to non-<blank>.                                                                    |  |
| 15445 | <b>File</b>                                                                                                    |  |
| 15446 | <i>Synopsis</i> :     f[ile][file]                                                                             |  |
| 15447 | If a <i>file</i> argument is specified, the alternate path name shall be set to the current path name, and     |  |
| 15448 | the current path name shall be set to <i>file</i> .                                                            |  |
| 15449 | Write an informational message. If the file has a current path name, it shall be included in this              |  |
| 15450 | message; otherwise, the message shall indicate that there is no current path name. If the edit                 |  |
| 15451 | buffer contains lines, the current line number and the number of lines in the edit buffer shall be             |  |
| 15452 | included in this message; otherwise, the message shall indicate that the edit buffer is empty. If              |  |
| 15453 | the edit buffer has been modified since the last complete write, this fact shall be included in this           |  |
| 15454 | message. If the <b>readonly</b> edit option is set, this fact shall be included in this message. The           |  |
| 15455 | message may contain other unspecified information.                                                             |  |
| 15456 | <i>Current line</i> : Unchanged.                                                                               |  |
| 15457 | <i>Current column</i> : Unchanged.                                                                             |  |

15458       **Global**

15459       *Synopsis:*     [2addr] g[lobal] /pattern/ [commands]  
 15460                   [2addr] v /pattern/ [commands]

15461       The optional '!' character after the **global** command shall be the same as executing the **v**  
 15462       command.

15463       If *pattern* is empty (for example, "//") or not specified, the last regular expression used in the  
 15464       editor command shall be used as the *pattern*. The *pattern* can be delimited by slashes (shown in  
 15465       the Synopsis), as well as any non-alphanumeric or non-<blank> character other than backslash,  
 15466       vertical line, double quote, or <newline>.

15467       If no lines are specified, the lines shall default to the entire file.

15468       The **global** and **v** commands are logically two-pass operations. First, mark the lines within the  
 15469       specified lines that match (**global**) or do not match (**v** or **global!**) the specified pattern. Second,  
 15470       execute the **ex** commands given by commands, with the current line ('.') set to each marked  
 15471       line. If an error occurs during this process, or the contents of the edit buffer are replaced (for  
 15472       example, by the **ex :edit** command) an error message shall be written and no more commands  
 15473       resulting from the execution of this command shall be processed.

15474       Multiple **ex** commands can be specified by entering multiple commands on a single line using a  
 15475       vertical line to delimit them, or one per line, by escaping each <newline> with a backslash.

15476       If no commands are specified:

- 15477       1. If in **ex** command mode, it shall be as if the **print** command were specified.
- 15478       2. Otherwise, no command shall be executed.

15479       For the **append**, **change**, and **insert** commands, the input text shall be included as part of the  
 15480       command, and the terminating period can be omitted if the command ends the list of  
 15481       commands. The **open** and **visual** commands can be specified as one of the commands, in which  
 15482       case each marked line shall cause the editor to enter open or visual mode. If open or visual mode  
 15483       is exited using the **vi Q** command, the current line shall be set to the next marked line, and open  
 15484       or visual mode reentered, until the list of marked lines is exhausted.

15485       The **global**, **v**, and **undo** commands cannot be used in commands. Marked lines may be deleted  
 15486       by commands executed for lines occurring earlier in the file than the marked lines. In this case,  
 15487       no commands shall be executed for the deleted lines.

15488       If the remembered search direction is not set, the **global** and **v** commands shall set it to forward.

15489       The **autoprint** and **autoindent** edit options shall be inhibited for the duration of the **g** or **v**  
 15490       command.

15491       *Current line:* If no commands executed, set to the last marked line. Otherwise, as specified for  
 15492       the executed **ex** commands.

15493       *Current column:* If no commands are executed, set to non-<blank>; otherwise, as specified for the  
 15494       individual **ex** commands.

15495 **Insert**15496 *Synopsis:*     [*laddr*] i[nsert][!]15497 Enter *ex* text input mode; the input text shall be placed before the specified line. If the line is zero  
15498 or 1, the text shall be placed at the beginning of the edit buffer.15499 This command shall be affected by the **number** and **autoindent** edit options; following the  
15500 command name with '!' shall cause the **autoindent** edit option setting to be toggled for the  
15501 duration of this command only.15502 *Current line:* Set to the last input line; if no lines were input, set to the line before the specified  
15503 line, or to the first line of the edit buffer if there are no lines preceding the specified line, or zero  
15504 if the edit buffer is empty.15505 *Current column:* Set to non-<blank>.15506 **Join**15507 *Synopsis:*     [*2addr*] j[oin][!][*count*][*flags*]15508 If *count* is specified:15509     If no address was specified, the **join** command shall behave as if *2addr* were the current line  
15510     and the current line plus *count* (.,. + *count*).15511     If one address was specified, the **join** command shall behave as if *2addr* were the specified  
15512     address and the specified address plus *count* (*addr*,*addr* + *count*).15513     If two addresses were specified, the **join** command shall behave as if an additional address,  
15514     equal to the last address plus *count* - 1 (*addr1*,*addr2*,*addr2* + *count* - 1), was specified.15515     If this would result in a second address greater than the last line of the edit buffer, it shall be  
15516     corrected to be equal to the last line of the edit buffer.15517 If no *count* is specified:15518     If no address was specified, the **join** command shall behave as if *2addr* were the current line  
15519     and the next line (.,. + 1).15520     If one address was specified, the **join** command shall behave as if *2addr* were the specified  
15521     address and the next line (*addr*,*addr* + 1).15522 Join the text from the specified lines together into a single line, which shall replace the specified  
15523 lines.15524 If a '!' character is appended to the command name, the **join** shall be without modification of  
15525 any line, independent of the current locale.15526 Otherwise, in the POSIX locale, set the current line to the first of the specified lines, and then, for  
15527 each subsequent line, proceed as follows:

- 15528     1. Discard leading <space>s from the line to be joined.
- 15529     2. If the line to be joined is now empty, delete it, and skip steps 3. through 5.
- 15530     3. If the current line ends in a <blank> character, or the first character of the line to be joined  
15531     is a ' ) ' character, join the lines without further modification.
- 15532     4. If the last character of the current line is a ' . ' , join the lines with two <space> characters  
15533     between them.

15534 5. Otherwise, join the lines with a single <space> character between them.

15535 *Current line*: Set to the first line specified.

15536 *Current column*: Set to non-<blank>.

## 15537 **List**

15538 *Synopsis*: `[2addr] l[ist][count][flags]`

15539 This command shall be equivalent to the *ex* command:

15540 `[2addr] p[rint][count] l[flags]`

15541 See **Print** on page 413.

## 15542 **Map**

15543 *Synopsis*: `map[!][lhs rhs]`

15544 If *lhs* and *rhs* are not specified:

- 15545 1. If '!' is specified, write the current list of text input mode maps.
- 15546 2. Otherwise, write the current list of command mode maps.
- 15547 3. Do nothing more.

15548 Implementations may restrict the set of characters accepted in *lhs* or *rhs*, except that printable  
 15549 characters and <blank> characters shall not be restricted. Additional restrictions shall be  
 15550 implementation-dependent. In both *lhs* and *rhs*, any character can be escaped with a  
 15551 <control>-V, in which case the character shall not be used to delimit *lhs* from *rhs*, and the  
 15552 escaping <control>-V shall be discarded.

15553 If the character '!' is appended to the **map** command name, the mapping shall be effective  
 15554 during open or visual text input mode rather than **open** or **visual** command mode. This allows  
 15555 *lhs* to have two different **map** definitions at the same time: one for command mode and one for  
 15556 text input mode.

15557 For command mode mappings:

15558 When the *lhs* is entered as any part of a *vi* command in open or visual mode (but not as part  
 15559 of the arguments to the command), the action shall be as if the corresponding *rhs* had been  
 15560 entered.

15561 If any character in the command, other than the first, is escaped using a <control>-V  
 15562 character, that character shall not be part of a match to an *lhs*.

15563 It is unspecified whether implementations shall support **map** commands where the *lhs* is  
 15564 more than a single character in length, where the first character of the *lhs* is printable.

15565 MAN If *lhs* contains more than one character and the first character is '#', followed by a sequence  
 15566 of digits corresponding to a numbered function key, then when this function key is typed it  
 15567 shall be mapped to *rhs*. Characters other than digits following a '#' character also  
 15568 represent the function key named by the characters in the *lhs* following the '#' and may be  
 15569 mapped to *rhs*. It is unspecified how function keys are named or what function keys are  
 15570 supported.

15571 For text input mode mappings:

|       |                                                                                                                                   |  |
|-------|-----------------------------------------------------------------------------------------------------------------------------------|--|
| 15572 | When the <i>lhs</i> is entered as any part of text entered in open or visual text input modes, the                                |  |
| 15573 | action shall be as if the corresponding <i>rhs</i> had been entered.                                                              |  |
| 15574 | If any character in the input text is escaped using a <control>-V character, that character shall                                 |  |
| 15575 | not be part of a match to an <i>lhs</i> .                                                                                         |  |
| 15576 | It is unspecified whether the <i>lhs</i> argument entered for the <b>map</b> or <b>unmap</b> commands is                          |  |
| 15577 | replaced in this fashion. Regardless of whether or not the replacement occurs, the effect of                                      |  |
| 15578 | the command shall be as if the replacement had not occurred.                                                                      |  |
| 15579 | If only part of the <i>lhs</i> is entered, it is unspecified how long the editor will wait for additional,                        |  |
| 15580 | possibly matching characters before treating the already entered characters as not matching the                                   |  |
| 15581 | <i>lhs</i> .                                                                                                                      |  |
| 15582 | The <i>rhs</i> characters shall themselves be subject to remapping, unless otherwise specified by the                             |  |
| 15583 | <b>remap</b> edit option, except that if the characters in <i>lhs</i> occur as prefix characters in <i>rhs</i> , those            |  |
| 15584 | characters shall not be remapped.                                                                                                 |  |
| 15585 | On block-mode terminals, the mapping need not occur immediately (for example, it may occur                                        |  |
| 15586 | after the terminal transmits a group of characters to the system), but it shall achieve the same                                  |  |
| 15587 | results as if it occurred immediately.                                                                                            |  |
| 15588 | <i>Current line</i> : Unchanged.                                                                                                  |  |
| 15589 | <i>Current column</i> : Unchanged.                                                                                                |  |
| 15590 | <b>Mark</b>                                                                                                                       |  |
| 15591 | <i>Synopsis:</i> [ <i>laddr</i> ] ma[ <i>rk</i> ] <i>character</i>                                                                |  |
| 15592 | [ <i>laddr</i> ] k <i>character</i>                                                                                               |  |
| 15593 | Implementations shall support <i>character</i> values of a single lowercase letter of the POSIX locale                            |  |
| 15594 | and the characters ' ' and ' ' '; support of other characters is implementation-dependent.                                        |  |
| 15595 | If executing the <b>vi m</b> command, set the specified mark to the current line and 1-based numbered                             |  |
| 15596 | character referenced by the current column, if any; otherwise, column position 1.                                                 |  |
| 15597 | Otherwise, set the specified mark to the specified line and 1-based numbered first non-<blank>                                    |  |
| 15598 | character in the line, if any; otherwise, the last character in the line, if any; otherwise, column                               |  |
| 15599 | position 1.                                                                                                                       |  |
| 15600 | The mark shall remain associated with the line until the mark is reset or the line is deleted. If a                               |  |
| 15601 | deleted line is restored by a subsequent <b>undo</b> command, any marks previously associated with                                |  |
| 15602 | the line, which have not been reset, shall be restored as well. Any use of a mark not associated                                  |  |
| 15603 | with a current line in the edit buffer shall be an error.                                                                         |  |
| 15604 | The marks ' and ' shall be set as described previously, immediately before the following events                                   |  |
| 15605 | occur in the editor:                                                                                                              |  |
| 15606 | 1. The use of ' \$ ' as an <i>ex</i> address                                                                                      |  |
| 15607 | 2. The use of a positive decimal number as an <i>ex</i> address                                                                   |  |
| 15608 | 3. The use of a search command as an <i>ex</i> address                                                                            |  |
| 15609 | 4. The use of a mark reference as an <i>ex</i> address                                                                            |  |
| 15610 | 5. The use of the following open and visual mode commands: <control>-[, %, (, ), [, ], {, }.                                      |  |
| 15611 | 6. The use of the following open and visual mode commands: ', <b>G</b> , <b>H</b> , <b>L</b> , <b>M</b> , <b>z</b> if the current |  |
| 15612 | line will change as a result of the command                                                                                       |  |

15613 7. The use of the open and visual mode commands: `/`, `?`, `N`, `'`, `n` if the current line or column  
15614 will change as a result of the command

15615 8. The use of the **ex** mode commands: **z**, **undo**, **global**, **v**

15616 For rules 1., 2., 3., and 4., the `'` and `'` marks shall not be set if the **ex** command is parsed as  
15617 specified by rule 6.a. in **Command Line Parsing in ex** on page 396.

15618 For rules 5., 6., and 7., the `'` and `'` marks shall not be set if the commands are used as motion  
15619 commands in open and visual mode.

15620 For rules 1., 2., 3., 4., 5., 6., 7., and 8., the `'` and `'` marks shall not be set if the command fails.

15621 The `'` and `'` marks shall be set as described previously, each time the contents of the edit buffer  
15622 are replaced (including the editing of the initial buffer), if in open or visual mode, or if in **ex**  
15623 mode and the edit buffer is not empty, before any commands or movements (including  
15624 commands or movements specified by the `-c` or `-t` options or the `+command` argument) are  
15625 executed on the edit buffer. If in open or visual mode, the marks shall be set as if executing the **vi**  
15626 **m** command; otherwise, as if executing the **ex mark** command.

15627 When changing from **ex** mode to open or visual mode, if the `'` and `'` marks are not already set,  
15628 the `'` and `'` marks shall be set as described previously.

15629 *Current line:* Unchanged.

15630 *Current column:* Unchanged.

## 15631 **Move**

15632 *Synopsis:* `[2addr] m[ove] laddr [flags]`

15633 Move the specified lines after the specified destination line. A destination of line zero specifies  
15634 that the lines shall be placed at the beginning of the edit buffer. It shall be an error if the  
15635 destination line is within the range of lines to be moved.

15636 *Current line:* Set to the last of the moved lines.

15637 *Current column:* Set to non-`<blank>`.

## 15638 **Next**

15639 *Synopsis:* `n[ext][!][+command][file ...]`

15640 If no `' ! '` is appended to the command name, and the edit buffer has been modified since the  
15641 last complete write, it shall be an error, unless the file is successfully written as specified by the  
15642 **autowrite** option.

15643 If one or more files is specified:

- 15644 1. Set the argument list to the specified file names.
- 15645 2. Set the current argument list reference to be the first entry in the argument list.
- 15646 3. Set the current path name to the first file name specified.

15647 Otherwise:

- 15648 1. It shall be an error if there are no more file names in the argument list after the file name  
15649 currently referenced.
- 15650 2. Set the current path name and the current argument list reference to the file name after the  
15651 file name currently referenced in the argument list.

15652 Replace the contents of the edit buffer with the contents of the file named by the current path  
15653 name. If for any reason the contents of the file cannot be accessed, the edit buffer shall be empty.

15654 This command shall be affected by the **autowrite** and **writany** edit options.

15655 The *+command* option shall be <blank> character-delimited; <blank> characters can be escaped  
15656 by preceding them with a backslash character. The *+command* shall be interpreted as an *ex*  
15657 command immediately after the contents of the edit buffer have been replaced and the current  
15658 line and column have been set.

15659 *Current line*: Set as described for the **edit** command.

15660 *Current column*: Set as described for the **edit** command.

## 15661 **Number**

15662 *Synopsis:*     [*2addr*] nu[mber][*count*][*flags*]  
15663               [*2addr*] #[*count*][*flags*]

15664 These commands shall be equivalent to the *ex* command:

15665 [*2addr*] p[rint][*count*] #[*flags*]

15666 See

## 15667 **Open**

15668 *Synopsis:*     [*1addr*] o[pen] /*pattern*/ [*flags*]

15669 This command need not be supported on block-mode terminals or terminals with insufficient  
15670 capabilities. If standard input, standard output, or standard error are not terminal devices, the  
15671 results are unspecified.

15672 Enter open mode.

15673 The trailing delimiter can be omitted from pattern at the end of the command line. If pattern is  
15674 empty (for example, " / / ") or not specified, the last regular expression used in the editor shall be  
15675 used as the pattern. The pattern can be delimited by slashes (shown in the Synopsis), as well as  
15676 any alphanumeric, or non-<blank> character other than backslash, vertical line, double quote, or  
15677 <newline> character.

15678 *Current line*: Set to the specified line.

15679 *Current column*: Set to non-<blank>.

## 15680 **Preserve**

15681 *Synopsis:*     pre[serve]

15682 Save the edit buffer in a form that can later be recovered by using the **-r** option or by using the *ex*  
15683 **recover** command. After the file has been preserved, a mail message shall be sent to the user.  
15684 This message shall be readable by invoking the *mailx* utility. The message shall contain the name  
15685 of the file, the time of preservation, and an *ex* command that could be used to recover the file.  
15686 Additional information may be included in the mail message.

15687 *Current line*: Unchanged.

15688 *Current column*: Unchanged.



15689      **Print**15690      *Synopsis:*      [*laddr*] p[rint][*count*][*flags*]15691      Write the addressed lines. The behavior is unspecified if the number of columns on the display is  
15692      less than the number of columns required to write any single character in the lines being written.15693      Non-printable characters, except for the <tab> character, shall be written as implementation-  
15694      dependent multi-character sequences.15695      If the # flag is specified or the **number** edit option is set, each line shall be preceded by its line  
15696      number in the following format:

15697      "%6dΔΔ", &lt;line number&gt;

15698      If the **l** flag is specified or the **list** edit option is set:15699      1. The characters listed in the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
15700      Table 5-1, Escape Sequences and Associated Actions shall be written as the corresponding  
15701      escape sequence.15702      2. Non-printable characters not in the System Interface Definitions volume of  
15703      IEEE Std. 1003.1-200x, Table 5-1, Escape Sequences and Associated Actions shall be written  
15704      as one three-digit octal number (with a preceding backslash) for each byte in the character  
15705      (most significant byte first). If the size of a byte on the system is greater than 9 bits, the  
15706      format used for non-printable characters is implementation-dependent.15707      3. The end of each line shall be marked with a '\$', and literal '\$' characters within the line  
15708      shall be written with a preceding backslash.15709      Long lines shall be folded; the length at which folding occurs is unspecified, but should be  
15710      appropriate for the output terminal, considering the number of columns of the terminal.15711      If a line is folded, and the **l** flag is not specified and the **list** edit option is not set, it is unspecified  
15712      whether a multi-column character at the folding position is separated; it shall not be discarded.15713      *Current line:* Set to the last written line.15714      *Current column:* Unchanged if the current line is unchanged; otherwise, set to non-<blank>.15715      **Put**15716      *Synopsis:*      [*laddr*] pu[t][*buffer*]15717      Append text from the specified buffer (by default, the unnamed buffer) to the specified line; line  
15718      zero specifies that the text shall be placed at the beginning of the edit buffer. Each portion of a  
15719      line in the buffer shall become a new line in the edit buffer, regardless of the mode of the buffer.15720      *Current line:* Set to the last line entered into the edit buffer.15721      *Current column:* Set to non-<blank>.15722      **Quit**15723      *Synopsis:*      q[uit][!]

15724      If no '!' is appended to the command name:

15725      1. If the edit buffer has been modified since the last complete write, it shall be an error.

15726      2. If there are file names in the argument list after the file name currently referenced, and the  
15727      last command was not a **quit**, **wq**, **xit**, or **ZZ** (see **Exit** on page 1064) command, it shall be  
15728      an error.

15729 Otherwise, terminate the editing session.

## 15730 **Read**

15731 *Synopsis:*     [*laddr*] r[ead][!][*file*]

15732 If '!' is not the first non-<blank> character to follow the command name, a copy of the  
15733 specified file shall be appended into the edit buffer after the specified line; line zero specifies that  
15734 the copy shall be placed at the beginning of the edit buffer. The number of lines and bytes read  
15735 shall be written. If no *file* is named, the current path name shall be the default. If there is no  
15736 current path name, then *file* shall become the current path name. If there is no current path name  
15737 or *file* operand, it shall be an error. Specifying a *file* that is not of type regular shall have  
15738 unspecified results.

15739 Otherwise, if *file* is preceded by '!', the rest of the line after the '!' shall have '%', '#', and  
15740 '!' characters expanded as described in **Command Line Parsing in ex** on page 396.

15741 The *ex* utility shall then pass two arguments to the program named by the *shell* edit option; the  
15742 first shall be -c and the second shall be the expanded arguments to the **read** command as a  
15743 single argument. The standard input of the program shall be set to the standard input of the *ex*  
15744 program when it was invoked. The standard error and standard output of the program shall be  
15745 appended into the edit buffer after the specified line.

15746 Each line in the copied file or program output (as delimited by <newline> characters or the end  
15747 of the file or output if it is not immediately preceded by a <newline> character), shall be a  
15748 separate line in the edit buffer. Any occurrences of <carriage-return> and <newline> character  
15749 pairs in the output shall be treated as single <newline> characters.

15750 The special meaning of the '!' following the **read** command can be overridden by escaping it  
15751 with a backslash character.

15752 *Current line:* If no lines are added to the edit buffer, unchanged. Otherwise, if in open or visual  
15753 mode, set to the first line entered into the edit buffer. Otherwise, set to the last line entered into  
15754 the edit buffer.

15755 *Current column:* Set to non-<blank>.

## 15756 **Recover**

15757 *Synopsis:*     rec[over][!] *file*

15758 If no '!' is appended to the command name, and the edit buffer has been modified since the  
15759 last complete write, it shall be an error.

15760 If no *file* operand is specified, then the current path name shall be used. If there is no current  
15761 path name or *file* operand, it shall be an error.

15762 If no recovery information has previously been saved about *file*, the **recover** command shall  
15763 behave identically to the **edit** command, and an informational message to this effect shall be  
15764 written.

15765 Otherwise, set the current path name to *file*, and replace the current contents of the edit buffer  
15766 with the recovered contents of *file*. If there are multiple instances of the file to be recovered, the  
15767 one most recently saved shall be recovered, and an informational message that there are  
15768 previous versions of the file that can be recovered shall be written. The editor shall behave as if  
15769 the contents of the edit buffer have already been modified.

15770 *Current file:* Set as described for the **edit** command.

|       |                                                                                                                |  |
|-------|----------------------------------------------------------------------------------------------------------------|--|
| 15771 | <i>Current column:</i> Set as described for the <b>edit</b> command.                                           |  |
| 15772 | <b>Rewind</b>                                                                                                  |  |
| 15773 | <i>Synopsis:</i> rew[ind][!]                                                                                   |  |
| 15774 | If no <b>'!'</b> is appended to the command name, and the edit buffer has been modified since the              |  |
| 15775 | last complete write, it shall be an error, unless the file is successfully written as specified by the         |  |
| 15776 | <b>autowrite</b> option.                                                                                       |  |
| 15777 | If the argument list is empty, it shall be an error.                                                           |  |
| 15778 | The current argument list reference and the current path name shall be set to the first file name              |  |
| 15779 | in the argument list.                                                                                          |  |
| 15780 | Replace the contents of the edit buffer with the contents of the file named by the current path                |  |
| 15781 | name. If for any reason the contents of the file cannot be accessed, the edit buffer shall be empty.           |  |
| 15782 | This command shall be affected by the <b>autowrite</b> and <b>writany</b> edit options.                        |  |
| 15783 | <i>Current line:</i> Set as described for the <b>edit</b> command.                                             |  |
| 15784 | <i>Current column:</i> Set as described for the <b>edit</b> command.                                           |  |
| 15785 | <b>Set</b>                                                                                                     |  |
| 15786 | <i>Synopsis:</i> se[t][option=[value]] ...[nooption ...][option? ...][all]                                     |  |
| 15787 | When no arguments are specified, write the value of the <b>term</b> edit option and those options              |  |
| 15788 | whose values have been changed from the default settings; when the argument <b>all</b> is specified,           |  |
| 15789 | write all of the option values.                                                                                |  |
| 15790 | Giving an option name followed by the character <b>'?'</b> shall cause the current value of that               |  |
| 15791 | option to be written. The <b>'?'</b> can be separated from the option name by zero or more <blank>             |  |
| 15792 | characters. The <b>'?'</b> shall be necessary only for Boolean valued options. Boolean options can be          |  |
| 15793 | given values by the form <b>set option</b> to turn them on or <b>set nooption</b> to turn them off; string and |  |
| 15794 | numeric options can be assigned by the form <b>set option=value</b> . Any <blank> characters in strings        |  |
| 15795 | can be included as is by preceding each <blank> with an escaping backslash. More than one                      |  |
| 15796 | option can be set or listed by a single set command by specifying multiple arguments, each                     |  |
| 15797 | separated from the next by one or more <blank> characters.                                                     |  |
| 15798 | See <b>Edit Options in ex</b> on page 425 for details about specific options.                                  |  |
| 15799 | <i>Current line:</i> Unchanged.                                                                                |  |
| 15800 | <i>Current column:</i> Unchanged.                                                                              |  |
| 15801 | <b>Shell</b>                                                                                                   |  |
| 15802 | <i>Synopsis:</i> sh[ell]                                                                                       |  |
| 15803 | Invoke the program named in the <b>shell</b> edit option with the single argument <b>-i</b> (interactive       |  |
| 15804 | mode). Editing shall be resumed when the program exits.                                                        |  |
| 15805 | <i>Current line:</i> Unchanged.                                                                                |  |
| 15806 | <i>Current column:</i> Unchanged.                                                                              |  |

|       |                                                                                                                                 |                                                                                                             |
|-------|---------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| 15807 | <b>Source</b>                                                                                                                   |                                                                                                             |
| 15808 | <i>Synopsis:</i>                                                                                                                | <code>so[urce] file</code>                                                                                  |
| 15809 | Read and execute <i>ex</i> commands from <i>file</i> . Lines in the file that contain no characters or only                     |                                                                                                             |
| 15810 | <blank> characters shall be ignored.                                                                                            |                                                                                                             |
| 15811 | <i>Current line:</i> As specified for the individual <i>ex</i> commands.                                                        |                                                                                                             |
| 15812 | <i>Current column:</i> As specified for the individual <i>ex</i> commands.                                                      |                                                                                                             |
| 15813 | <b>Substitute</b>                                                                                                               |                                                                                                             |
| 15814 | <i>Synopsis:</i>                                                                                                                | <code>[2addr] s[substitute][/<i>pattern</i>/<i>repl</i>][<i>options</i>][<i>count</i>][<i>flags</i>]</code> |
| 15815 |                                                                                                                                 | <code>[2addr] &amp;[<i>options</i>][<i>count</i>][<i>flags</i>]</code>                                      |
| 15816 |                                                                                                                                 | <code>[2addr] ~[<i>options</i>][<i>count</i>][<i>flags</i>]</code>                                          |
| 15817 | Replace the first instance of the pattern <i>pattern</i> by the string <i>repl</i> on each specified line. (See                 |                                                                                                             |
| 15818 | <b>Regular Expressions in ex</b> on page 424 and <b>Replacement Strings in ex</b> on page 425.) Any non-                        |                                                                                                             |
| 15819 | alphabetic, non-<blank> delimiter other than ' <code>\</code> ', ' <code> </code> ', double quote, or <newline> character       |                                                                                                             |
| 15820 | can be used instead of ' <code>/</code> '. Backslash characters can be used to escape delimiters, backslash                     |                                                                                                             |
| 15821 | characters, and other special characters.                                                                                       |                                                                                                             |
| 15822 | The trailing delimiter can be omitted from <i>pattern</i> or from <i>repl</i> at the end of the command line. If                |                                                                                                             |
| 15823 | both <i>pattern</i> and <i>repl</i> are not specified or are empty (for example, " <code>/</code> "), the last <b>s</b> command |                                                                                                             |
| 15824 | shall be repeated. If only <i>pattern</i> is not specified or is empty, the last regular expression used in                     |                                                                                                             |
| 15825 | the editor shall be used as the pattern. If only <i>repl</i> is not specified or is empty, the pattern shall be                 |                                                                                                             |
| 15826 | replaced by nothing. If the entire replacement pattern is ' <code>%</code> ', the last replacement pattern to an                |                                                                                                             |
| 15827 | <b>s</b> command shall be used.                                                                                                 |                                                                                                             |
| 15828 | Entering a <carriage-return> in <i>repl</i> (which requires an escaping backslash in <i>ex</i> mode and an                      |                                                                                                             |
| 15829 | escaping <control>-V in open or <i>vi</i> mode) shall split the line at that point, creating a new line in                      |                                                                                                             |
| 15830 | the edit buffer. The <carriage-return> shall be discarded.                                                                      |                                                                                                             |
| 15831 | If options include the letter ' <code>g</code> ' ( <b>global</b> ), all non-overlapping instances of the pattern in the line    |                                                                                                             |
| 15832 | shall be replaced.                                                                                                              |                                                                                                             |
| 15833 | If options includes the letter ' <code>c</code> ' ( <b>confirm</b> ), then before each substitution the line shall be           |                                                                                                             |
| 15834 | written; the written line shall reflect all previous substitutions. On the following line, <space>                              |                                                                                                             |
| 15835 | characters shall be written beneath the characters from the line that are before the <i>pattern</i> to be                       |                                                                                                             |
| 15836 | replaced, and ' <code>^</code> ' characters written beneath the characters included in the <i>pattern</i> to be                 |                                                                                                             |
| 15837 | replaced. The <i>ex</i> utility shall then wait for a response from the user. An affirmative response                           |                                                                                                             |
| 15838 | shall cause the substitution to be done, while any other input shall not make the substitution. An                              |                                                                                                             |
| 15839 | affirmative response shall consist of a line with the affirmative response (as defined by the                                   |                                                                                                             |
| 15840 | current locale) at the beginning of the line. This line shall be subject to editing in the same way as                          |                                                                                                             |
| 15841 | the <i>ex</i> command line.                                                                                                     |                                                                                                             |
| 15842 | If interrupted (see the ASYNCHRONOUS EVENTS section), any modifications confirmed by the                                        |                                                                                                             |
| 15843 | user shall be preserved in the edit buffer after the interrupt.                                                                 |                                                                                                             |
| 15844 | If the remembered search direction is not set, the <b>s</b> command shall set it to forward.                                    |                                                                                                             |
| 15845 | In the second Synopsis, the <b>&amp;</b> command shall repeat the previous substitution, as if the <b>&amp;</b>                 |                                                                                                             |
| 15846 | command were replaced by:                                                                                                       |                                                                                                             |
| 15847 | <code>s/<i>pattern</i>/<i>repl</i>/</code>                                                                                      |                                                                                                             |
| 15848 | where <i>pattern</i> and <i>repl</i> are as specified in the previous <b>s</b> , <b>&amp;</b> , or <b>~</b> command.            |                                                                                                             |

15849 In the third Synopsis, the `~` command shall repeat the previous substitution, as if the `'~'` were  
 15850 replaced by:

15851 `s/pattern/repl/`

15852 where *pattern* shall be the last regular expression specified to the editor, and *repl* shall be from  
 15853 the previous substitution (including `&` and `~`) command.

15854 These commands shall be affected by the `LC_MESSAGES` environment variable.

15855 *Current line*: Set to the last line in which a substitution occurred, or, unchanged if no  
 15856 substitution occurred.

15857 *Current column*: Set to non-`<blank>`.

## 15858 Suspend

15859 *Synopsis*: `su[suspend][!]`  
 15860 `st[op][!]`

15861 Allow control to return to the invoking process; `ex` shall suspend itself as if it had received the  
 15862 SIGTSTP signal. The suspension shall occur only if job control is enabled in the invoking shell  
 15863 (see the description of `set -m`).

15864 These commands shall be affected by the **autowrite** and **writeln** edit options.

15865 The current **susp** character (see `stty`) shall have the same affect as the **suspend** command.

## 15866 Tag

15867 *Synopsis*: `ta[g][!] tagstring`

15868 The results are unspecified if the format of a tags file is not as specified by the `ctags` utility (see  
 15869 `ctags`) description.

15870 The **tag** command shall search for *tagstring* in the tag files referred to by the **tag** edit option, in  
 15871 the order they are specified, until a reference to *tagstring* is found. Files shall be searched from  
 15872 beginning to end. If no reference is found, it shall be an error and an error message to this effect  
 15873 shall be written. If the reference is not found, or if an error occurs while processing a file referred  
 15874 to in the **tag** edit option, it shall be an error, and an error message shall be written at the first  
 15875 occurrence of such an error.

15876 Otherwise, if the tags file contained a pattern, the pattern shall be treated as a regular expression  
 15877 used in the editor; for example, for the purposes of the **s** command.

15878 If the *tagstring* is in a file with a different name than the current path name, set the current path  
 15879 name to the name of that file, and replace the contents of the edit buffer with the contents of that  
 15880 file. In this case, if no `'!'` is appended to the command name, and the edit buffer has been  
 15881 modified since the last complete write, it shall be an error, unless the file is successfully written  
 15882 as specified by the **autowrite** option.

15883 This command shall be affected by the **autowrite**, **tag**, **taglength**, and **writeln** edit options.

15884 *Current line*: If the tags file contained a line number, set to that line number. If the line number is  
 15885 larger than the last line in the edit buffer, an error message shall be written and the current line  
 15886 shall be set as specified for the **edit** command.

15887 If the tags file contained a pattern, set to the first occurrence of the pattern in the file. If no  
 15888 matching pattern is found, an error message shall be written and the current line shall be set as  
 15889 specified for the **edit** command.

15890 *Current column:* If the tags file contained a line-number reference and that line-number was not  
 15891 larger than the last line in the edit buffer, or if the tags file contained a pattern and that pattern  
 15892 was found, set to non-<blank>. Otherwise, set as specified for the **edit** command.

### 15893 **Unabbreviate**

15894 *Synopsis:*     una[bbrev] *lhs*

15895 If *lhs* is not an entry in the current list of abbreviations (see **Abbreviate** on page 403), it shall be  
 15896 an error. Otherwise, delete *lhs* from the list of abbreviations.

15897 *Current line:* Unchanged.

15898 *Current column:* Unchanged.

### 15899 **Undo**

15900 *Synopsis:*     u[ndo]

15901 Reverse the changes made by the last command that modified the contents of the edit buffer,  
 15902 including **undo**. For this purpose, the **global**, **v**, **open**, and **visual** commands, and commands  
 15903 resulting from buffer executions and mapped character expansions, are considered single  
 15904 commands.

15905 If no action that can be undone preceded the **undo** command, it shall be an error.

15906 If the **undo** command restores lines that were marked, the mark shall also be restored unless it  
 15907 was reset subsequent to the deletion of the lines.

15908 *Current line:*

- 15909     1. If lines are added or changed in the file, set to the first line added or changed.
- 15910     2. Set to the line before the first line deleted, if it exists.
- 15911     3. Set to 1 if the edit buffer is not empty.
- 15912     4. Set to zero.

15913 *Current column:* Set to non-<blank>.

### 15914 **Unmap**

15915 *Synopsis:*     unm[ap][!] *lhs*

15916 If '!' is appended to the command name, and if *lhs* is not an entry in the list of text input mode  
 15917 map definitions, it shall be an error. Otherwise, delete *lhs* from the list of text input mode map  
 15918 definitions.

15919 If no '!' is appended to the command name, and if *lhs* is not an entry in the list of command  
 15920 mode map definitions, it shall be an error. Otherwise, delete *lhs* from the list of command mode  
 15921 map definitions.

15922 *Current line:* Unchanged.

15923 *Current column:* Unchanged.

|       |                                                                                                                           |                                                   |
|-------|---------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|
| 15924 | <b>Version</b>                                                                                                            |                                                   |
| 15925 | <i>Synopsis:</i>                                                                                                          | <code>ve[rsion]</code>                            |
| 15926 | Write a message containing version information for the editor. The format of the message is                               |                                                   |
| 15927 | unspecified.                                                                                                              |                                                   |
| 15928 | <i>Current line:</i>                                                                                                      | Unchanged.                                        |
| 15929 | <i>Current column:</i>                                                                                                    | Unchanged.                                        |
| 15930 | <b>Visual</b>                                                                                                             |                                                   |
| 15931 | <i>Synopsis:</i>                                                                                                          | <code>[laddr] vi[sual][type][count][flags]</code> |
| 15932 | If <b>ex</b> is currently in open or visual mode, the Synopsis and behavior of the visual command shall                   |                                                   |
| 15933 | be the same as the <b>edit</b> command, as specified by <b>Edit</b> on page 405.                                          |                                                   |
| 15934 | Otherwise, this command need not be supported on block-mode terminals or terminals with                                   |                                                   |
| 15935 | insufficient capabilities. If standard input, standard output, or standard error are not terminal                         |                                                   |
| 15936 | devices, the results are unspecified.                                                                                     |                                                   |
| 15937 | If <i>count</i> is specified, the value of the <b>window</b> edit option shall be set to <i>count</i> (as described in    |                                                   |
| 15938 | <b>window</b> on page 432). If the '^' type character was also specified, the <b>window</b> edit option shall             |                                                   |
| 15939 | be set before being used by the type character.                                                                           |                                                   |
| 15940 | Enter visual mode. If <i>type</i> is not specified, it shall be as if a <i>type</i> of '+' was specified. The <i>type</i> |                                                   |
| 15941 | shall cause the following effects:                                                                                        |                                                   |
| 15942 | + Place the beginning of the specified line at the top of the display.                                                    |                                                   |
| 15943 | - Place the end of the specified line at the bottom of the display.                                                       |                                                   |
| 15944 | . Place the beginning of the specified line in the middle of the display.                                                 |                                                   |
| 15945 | ^ If the specified line is less than or equal to the value of the <b>window</b> edit option, set the line                 |                                                   |
| 15946 | to 1; otherwise, decrement the line by the value of the <b>window</b> edit option minus 1. Place                          |                                                   |
| 15947 | the beginning of this line as close to the bottom of the displayed lines as possible, while still                         |                                                   |
| 15948 | displaying the value of the <b>window</b> edit option number of lines.                                                    |                                                   |
| 15949 | <i>Current line:</i>                                                                                                      | Set to the specified line.                        |
| 15950 | <i>Current column:</i>                                                                                                    | Set to non-<blank>.                               |
| 15951 | <b>Write</b>                                                                                                              |                                                   |
| 15952 | <i>Synopsis:</i>                                                                                                          | <code>[2addr] w[rite][!][&gt;&gt;][file]</code>   |
| 15953 |                                                                                                                           | <code>[2addr] w[rite][!][file]</code>             |
| 15954 |                                                                                                                           | <code>[2addr] wq[!][&gt;&gt;][file]</code>        |
| 15955 | If no lines are specified, the lines shall default to the entire file.                                                    |                                                   |
| 15956 | The command <b>wq</b> shall be equivalent to a <b>write</b> command followed by a <b>quit</b> command; <b>wq!</b>         |                                                   |
| 15957 | shall be equivalent to <b>write!</b> followed by <b>quit</b> . In both cases, if the <b>write</b> command fails, the      |                                                   |
| 15958 | <b>quit</b> shall not be attempted.                                                                                       |                                                   |
| 15959 | If the command name is not followed by one or more <blank> characters, or <i>file</i> is not preceded                     |                                                   |
| 15960 | by a '!' character, the <b>write</b> shall be to a file.                                                                  |                                                   |
| 15961 | 1. If the >> argument is specified, and the file already exists, the lines shall be appended to                           |                                                   |
| 15962 | the file instead of replacing its contents. If the >> argument is specified, and the file does                            |                                                   |
| 15963 | not already exist, it is unspecified whether the write shall proceed as if the >> argument                                |                                                   |

|       |                                                                                                                     |  |
|-------|---------------------------------------------------------------------------------------------------------------------|--|
| 15964 | had not been specified or if the write shall fail.                                                                  |  |
| 15965 | 2. If the <b>readonly</b> edit option is set (see <b>readonly</b> on page 429), the <b>write</b> shall fail.        |  |
| 15966 | 3. If <i>file</i> is specified, and is not the current path name, and the file exists, the <b>write</b> shall fail. |  |
| 15967 | 4. If <i>file</i> is not specified, the current path name shall be used. If there is no current path name,          |  |
| 15968 | the <b>write</b> command shall fail.                                                                                |  |
| 15969 | 5. If the current path name is used, and the current path name has been changed by the <b>file</b>                  |  |
| 15970 | or <b>read</b> commands, and the file exists, the <b>write</b> shall fail. If the <b>write</b> is successful,       |  |
| 15971 | subsequent <b>writes</b> shall not fail for this reason (unless the current path name is changed                    |  |
| 15972 | again).                                                                                                             |  |
| 15973 | 6. If the whole edit buffer is not being written, and the file to be written exists, the <b>write</b> shall         |  |
| 15974 | fail.                                                                                                               |  |
| 15975 | For rules 1., 2., 4., and 5., the <b>write</b> can be forced by appending the character '!' to the                  |  |
| 15976 | command name.                                                                                                       |  |
| 15977 | For rules 2., 4., and 5., the <b>write</b> can be forced by setting the <b>writeany</b> edit option.                |  |
| 15978 | Additional, implementation-dependent tests may cause the <b>write</b> to fail.                                      |  |
| 15979 | If the edit buffer is empty, a file without any contents shall be written.                                          |  |
| 15980 | An informational message shall be written noting the number of lines and bytes written.                             |  |
| 15981 | Otherwise, if the command is followed by one or more <blank> characters, and file is preceded                       |  |
| 15982 | by '!', the rest of the line after the '!' shall have '%', '#', and '!' characters expanded as                      |  |
| 15983 | described in <b>Command Line Parsing in ex</b> on page 396.                                                         |  |
| 15984 | The <b>ex</b> utility shall then pass two arguments to the program named by the <b>shell</b> edit option; the       |  |
| 15985 | first shall be -c and the second shall be the expanded arguments to the <b>write</b> command as a                   |  |
| 15986 | single argument. The specified lines shall be written to the standard input of the command. The                     |  |
| 15987 | standard error and standard output of the program, if any, shall be written as described for the                    |  |
| 15988 | <b>print</b> command. If the last character in that output is not a <newline> character, a <newline>                |  |
| 15989 | shall be written at the end of the output.                                                                          |  |
| 15990 | The special meaning of the '!' following the <b>write</b> command can be overridden by escaping it                  |  |
| 15991 | with a backslash character.                                                                                         |  |
| 15992 | <i>Current line:</i> Unchanged.                                                                                     |  |
| 15993 | <i>Current column:</i> Unchanged.                                                                                   |  |
| 15994 | <b>Write and Exit</b>                                                                                               |  |
| 15995 | <i>Synopsis:</i> [2addr] x[it][!][file]                                                                             |  |
| 15996 | If the edit buffer has not been modified since the last complete <b>write</b> , <b>xit</b> shall be equivalent to   |  |
| 15997 | the <b>quit</b> command, or if a '!' is appended to the command name, to <b>quit!</b> .                             |  |
| 15998 | Otherwise, <b>xit</b> shall be equivalent to the <b>wq</b> command, or if a '!' is appended to the command          |  |
| 15999 | name, to <b>wq!</b> .                                                                                               |  |
| 16000 | <i>Current line:</i> Unchanged.                                                                                     |  |
| 16001 | <i>Current line:</i> Unchanged.                                                                                     |  |



|       |                                                                                                                         |                                                   |
|-------|-------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------|
| 16002 | <b>Yank</b>                                                                                                             |                                                   |
| 16003 | <i>Synopsis:</i>                                                                                                        | <code>[laddr] ya[nk][buffer][count]</code>        |
| 16004 | Copy the specified lines to the specified buffer (by default, the unnamed buffer), which shall                          |                                                   |
| 16005 | become a line-mode buffer.                                                                                              |                                                   |
| 16006 | <i>Current line:</i> Unchanged.                                                                                         |                                                   |
| 16007 | <i>Current line:</i> Unchanged.                                                                                         |                                                   |
| 16008 | <b>Adjust Window</b>                                                                                                    |                                                   |
| 16009 | <i>Synopsis:</i>                                                                                                        | <code>[laddr] z[!][type ...][count][flags]</code> |
| 16010 | If no line is specified, the current line shall be the default; if <i>type</i> is omitted as well, the current          |                                                   |
| 16011 | line value shall first be incremented by 1. If incrementing the current line would cause it to be                       |                                                   |
| 16012 | greater than the last line in the edit buffer, it shall be an error.                                                    |                                                   |
| 16013 | If there are <blank> characters between the <i>type</i> argument and the preceding <i>z</i> command name                |                                                   |
| 16014 | or optional '!' character, it shall be an error.                                                                        |                                                   |
| 16015 | If <i>count</i> is specified, the value of the <b>window</b> edit option shall be set to <i>count</i> (as described in  |                                                   |
| 16016 | <b>window</b> on page 432). If <i>count</i> is omitted, it shall default to 2 times the value of the <b>scroll</b> edit |                                                   |
| 16017 | option, or if ! was specified, the number of lines in the display minus 1.                                              |                                                   |
| 16018 | If <i>type</i> is omitted, then <i>count</i> lines starting with the specified line shall be written. Otherwise,        |                                                   |
| 16019 | <i>count</i> lines starting with the line specified by the <i>type</i> argument shall be written.                       |                                                   |
| 16020 | The <i>type</i> argument shall change the lines to be written. The possible values of <i>type</i> are as follows:       |                                                   |
| 16021 | – The specified line shall be decremented by the following value:                                                       |                                                   |
| 16022 | $((\text{number of ``-'' characters}) \text{ W } count) - 1)$                                                           |                                                   |
| 16023 | If the calculation would result in a number less than 1, it shall be an error. Write lines from                         |                                                   |
| 16024 | the edit buffer, starting at the new value of line, until <i>count</i> lines or the last line in the edit               |                                                   |
| 16025 | buffer has been written.                                                                                                |                                                   |
| 16026 | + The specified line shall be incremented by the following value:                                                       |                                                   |
| 16027 | $((\text{number of ``+'' characters}) - 1) \text{ W } count) + 1)$                                                      |                                                   |
| 16028 | If the calculation would result in a number greater than the last line in the edit buffer, it                           |                                                   |
| 16029 | shall be an error. Write lines from the edit buffer, starting at the new value of line, until                           |                                                   |
| 16030 | <i>count</i> lines or the last line in the edit buffer has been written.                                                |                                                   |
| 16031 | =, . If more than a single ' . ' or '= ' is specified, it shall be an error. The following steps shall be               |                                                   |
| 16032 | taken:                                                                                                                  |                                                   |
| 16033 | 1. If <i>count</i> is zero, nothing shall be written.                                                                   |                                                   |
| 16034 | 2. Write as many of the <i>N</i> lines before the current line in the edit buffer as exist. If <i>count</i>             |                                                   |
| 16035 | or '!' was specified, <i>N</i> shall be:                                                                                |                                                   |
| 16036 | $(count - 1) / 2$                                                                                                       |                                                   |
| 16037 | Otherwise, <i>N</i> shall be:                                                                                           |                                                   |
| 16038 | $(count - 3) / 2$                                                                                                       |                                                   |
| 16039 | If <i>N</i> is a number less than 3, no lines shall be written.                                                         |                                                   |

16040 3. If '=' was specified as the type character, write a line consisting of the smaller of the  
 16041 number of columns in the display divided by two, or 40 '-' characters.

16042 4. Write the current line.

16043 5. Repeat step 3.

16044 6. Write as many of the *N* lines after the current line in the edit buffer as exist. *N* shall be  
 16045 defined as in step 2. If *N* is a number less than 3, no lines shall be written. current line  
 16046 in the edit buffer as exist. If count is less than 3, no lines shall be written.

16047 ^ The specified line shall be decremented by the following value:  
 16048 ((number of '^' characters) +1) W count) -1

16049 If the calculation would result in a number less than 1, it shall be an error. Write lines from  
 16050 the edit buffer, starting at the new value of line, until *count* lines or the last line in the edit  
 16051 buffer has been written.

16052 *Current line:* Set to the last line written, unless the type is =, in which case, set to the specified  
 16053 line.

16054 *Current column:* Set to non-<blank>.

16055 **Escape**

16056 *Synopsis:* ! *command*  
 16057 [addr]! *command*

16058 The contents of the line after the '!' shall have '%', '#', and '!' characters expanded as  
 16059 described in **Command Line Parsing in ex** on page 396. If the expansion causes the text of the  
 16060 line to change, it shall be redisplayed, preceded by a single '!' character.

16061 The **ex** utility shall execute the program named by the **shell** edit option. It shall pass two  
 16062 arguments to the program; the first shall be -c, and the second shall be the expanded arguments  
 16063 to the ! command as a single argument.

16064 If no lines are specified, the standard input, standard output, and standard error of the program  
 16065 shall be set to the standard input, standard output, and standard error of the **ex** program when it  
 16066 was invoked. In addition, a warning message shall be written if the edit buffer has been  
 16067 modified since the last complete write, and the **warn** edit option is set.

16068 If lines are specified, they shall be passed to the program as standard input, and the standard  
 16069 output and standard error of the program shall replace those lines in the edit buffer. Each line in  
 16070 the program output (as delimited by <newline> characters or the end of the output if it is not  
 16071 immediately preceded by a <newline> character), shall be a separate line in the edit buffer. Any  
 16072 occurrences of <carriage-return> and <newline> character pairs in the output shall be treated as  
 16073 single <newline> characters. The specified lines shall be copied into the unnamed buffer before  
 16074 they are replaced, and the unnamed buffer shall become a line-mode buffer.

16075 If in **ex** mode, a single '!' character shall be written when the program completes.

16076 This command shall be affected by the **shell** and **warn** edit options. If no lines are specified, this  
 16077 command shall be affected by the **autowrite** and **writany** edit options. If lines are specified, this  
 16078 command shall be affected by the **autoprint** edit option.

16079 *Current line:*

16080 1. If no lines are specified, unchanged.

|       |                                                                                                                     |  |
|-------|---------------------------------------------------------------------------------------------------------------------|--|
| 16081 | 2. Otherwise, set to the last line read in, if any lines are read in.                                               |  |
| 16082 | 3. Otherwise, set to the line before the first line of the lines specified, if that line exists.                    |  |
| 16083 | 4. Otherwise, set to the first line of the edit buffer if the edit buffer is not empty.                             |  |
| 16084 | 5. Otherwise, set to zero.                                                                                          |  |
| 16085 | <i>Current column:</i> If no lines are specified, unchanged. Otherwise, set to non-<blank>.                         |  |
| 16086 | <b>Shift Left</b>                                                                                                   |  |
| 16087 | <i>Synopsis:</i> [2addr] <[< ... ][count][flags]                                                                    |  |
| 16088 | Shift the specified lines to the start of the line; the number of column positions to be shifted shall              |  |
| 16089 | be the number of command characters times the value of the <b>shiftwidth</b> edit option. Only                      |  |
| 16090 | leading <blank> characters shall be deleted or changed into other <blank> characters in shifting;                   |  |
| 16091 | other characters shall not be affected.                                                                             |  |
| 16092 | Lines to be shifted shall be copied into the unnamed buffer, which shall become a line-mode                         |  |
| 16093 | buffer.                                                                                                             |  |
| 16094 | This command shall be affected by the <b>autoprint</b> edit option.                                                 |  |
| 16095 | <i>Current line:</i> Set to the last line in the lines specified.                                                   |  |
| 16096 | <i>Current column:</i> Set to non-<blank>.                                                                          |  |
| 16097 | <b>Shift Right</b>                                                                                                  |  |
| 16098 | <i>Synopsis:</i> [2addr] >[> ... ][count][flags]                                                                    |  |
| 16099 | Shift the specified lines away from the start of the line; the number of column positions to be                     |  |
| 16100 | shifted shall be the number of command characters times the value of the <b>shiftwidth</b> edit option.             |  |
| 16101 | The shift shall be accomplished by adding <blank> characters as a prefix to the line or changing                    |  |
| 16102 | leading <blank> characters into other <blank> characters. Empty lines shall not be changed.                         |  |
| 16103 | Lines to be shifted shall be copied into the unnamed buffer, which shall become a line-mode                         |  |
| 16104 | buffer.                                                                                                             |  |
| 16105 | This command shall be affected by the <b>autoprint</b> edit option.                                                 |  |
| 16106 | <i>Current line:</i> Set to the last line in the lines specified.                                                   |  |
| 16107 | <i>Current column:</i> Set to non-<blank>.                                                                          |  |
| 16108 | <b>&lt;control&gt;-D</b>                                                                                            |  |
| 16109 | <i>Synopsis:</i> <control>-D                                                                                        |  |
| 16110 | Write the next <i>n</i> lines, where <i>n</i> is the minimum of the values of the <b>scroll</b> edit option and the |  |
| 16111 | number of lines after the current line in the edit buffer. If the current line is the last line of the              |  |
| 16112 | edit buffer it shall be an error.                                                                                   |  |
| 16113 | <i>Current line:</i> Set to the last line written.                                                                  |  |
| 16114 | <i>Current column:</i> Set to non-<blank>.                                                                          |  |

**Write Line Number**

*Synopsis:*     [*laddr*] = [*flags*]

If *line* is not specified, it shall default to the last line in the edit buffer. Write the line number of the specified line.

*Current line:* Unchanged.

*Current column:* Unchanged.

**Execute**

*Synopsis:*     [*2addr*] @ *buffer*

[*2addr*] \* *buffer*

If no buffer is specified or is specified as '@' or '\*', the last buffer executed shall be used. If no previous buffer has been executed, it shall be an error.

For each line specified by the addresses, set the current line ('.') to the specified line, and execute the contents of the named *buffer* (as they were at the time the @ command was executed) as *ex* commands. For each line of a line-mode buffer, and all but the last line of a character-mode buffer, the *ex* command parser shall behave as if the line was terminated by a <newline> character.

If an error occurs during this process, or a line specified by the addresses does not exist when the current line would be set to it, or more than a single line was specified by the addresses, and the contents of the edit buffer are replaced (for example, by the *ex :edit* command) an error message shall be written, and no more commands resulting from the execution of this command shall be processed.

*Current line:* As specified for the individual *ex* commands.

*Current column:* As specified for the individual *ex* commands.

**Regular Expressions in ex**

The *ex* utility shall support regular expressions that are a superset of the basic regular expressions described in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 9.3, Basic Regular Expressions. A null regular expression ("/") shall be equivalent to the last regular expression encountered.

Regular expressions can be used in addresses to specify lines and, in some commands (for example, the **substitute** command), to specify portions of a line to be substituted.

The following constructs can be used to enhance the basic regular expressions:

\< Match the beginning of a *word*. (See the definition of *word* at the beginning of **Command Descriptions in ex** on page 401.)

\> Match the end of a *word*.

~ Match the replacement part of the last **substitute** command. The tilde ('~') character can be escaped in a regular expression to become a normal character with no special meaning. The backslash shall be discarded.

When the editor option **magic** is not set, the only characters with special meanings shall be '^' at the beginning of a pattern, '\$' at the end of a pattern, and '\'. The characters '.', '\*', '[', and '~' shall be treated as ordinary characters unless preceded by a '\'; when preceded by a '\' they shall regain their special meaning, or in the case of backslash, be handled as a single backslash. Backslashes used to escape other characters shall be discarded.

**Replacement Strings in ex**

The character `'&'` (`'\&'` if the editor option **magic** is not set) in the replacement string shall stand for the text matched by the pattern to be replaced. The character `'~'` (`'\~'` if **magic** is not set) shall be replaced by the replacement part of the previous **substitute** command. The sequence `'\n'`, where *n* is an integer, shall be replaced by the text matched by the pattern enclosed in the *n*th set of parentheses `'\('` and `'\).'`.

The strings `'\l'`, `'\u'`, `'\L'`, and `'\U'` can be used to modify the case of elements in the replacement string (using the `'\&'` or `"\"`digit) notation. The string `'\l'` (`'\u'`) shall cause the character that follows to be converted to lowercase (uppercase). The string `'\L'` (`'\U'`) shall cause all characters subsequent to it to be converted to lowercase (uppercase) as they are inserted by the substitution until the string `'\e'` or `'\E'`, or the end of the replacement string, is encountered.

Otherwise, any character following a backslash shall be treated as that literal character, and the escaping backslash shall be discarded.

An example of case conversion with the **s** command is as follows:

```
:p
The cat sat on the mat.
:s/\<.at\>/\u&/gp
The Cat Sat on the Mat.
:s/S\(.*\)M/S\U\1\eM/p
The Cat SAT ON THE Mat.
```

**Edit Options in ex**

The **ex** utility has a number of options that modify its behavior. These options have default settings, which can be changed using the **set** command.

Options are Boolean unless otherwise specified.

**autoindent, ai**

[Default *unset*]

If **autoindent** is set, each line in input mode shall be indented (using first as many `<tab>` characters as possible, as determined by the editor option **tabstop**, and then using `<space>` characters) to align with another line, as follows:

1. If in open or visual mode and the text input is part of a line-oriented command (see the EXTENDED DESCRIPTION in *vi*), align to the first column. Otherwise, if in open or visual mode, indentation for each line shall be set as follows:
  - a. If a line was previously inserted as part of this command, it shall be set to the indentation of the last inserted line by default, or as otherwise specified for the `<control>` character in `<REFERENCE UNDEFINED>(vinputctld)`.
  - b. Otherwise, it shall be set to the indentation of the previous current line, if any; otherwise, to the first column.
2. For the **ex** **a**, **i**, and **c** commands, indentation for each line shall be set as follows:
  - a. If a line was previously inserted as part of this command, it shall be set to the indentation of the last inserted line by default, or as otherwise specified for the *eof* character in **Scroll** on page 400.

- 16199           b. Otherwise, if the command is the **ex a** command, it shall be set to the line appended  
16200           after, if any; otherwise to the first column.
- 16201           c. Otherwise, if the command is the **ex i** command, it shall be set to the line inserted  
16202           before, if any; otherwise to the first column.
- 16203           d. Otherwise, if the command is the **ex c** command, it shall be set to the indentation of  
16204           the line replaced.

#### 16205       **autoprint, ap**

16206       [Default *set*]

16207       If **autoprint** is set, the current line shall be written after each **ex** command that modifies the  
16208       contents of the current edit buffer, and after each **tag** command for which the tag search pattern  
16209       was found or tag line number was valid, unless:

- 16210       1. The command was executed while in open or visual mode.
- 16211       2. The command was executed as part of a **global** or **v** command or @ buffer execution.
- 16212       3. The command was the form of the **read** command that reads a file into the edit buffer.
- 16213       4. The command was the **append**, **change**, or **insert** command.
- 16214       5. The command was not terminated by a <newline> character.
- 16215       6. The current line shall be written by a flag specified to the command; for example, **delete #**  
16216       shall write the current line as specified for the flag modifier to the **delete** command, and  
16217       not as specified by the **autoprint** edit option.

#### 16218       **autowrite, aw**

16219       [Default *unset*]

16220       If **autowrite** is set, and the edit buffer has been modified since it was last completely written to  
16221       any file, the contents of the edit buffer shall be written as if the **ex write** command had been  
16222       specified without arguments, before each command affected by the **autowrite** edit option is  
16223       executed. Appending the character '!' to the command name of any of the **ex** commands  
16224       except '!' shall prevent the write. If the write fails, it shall be an error and the command shall  
16225       not be executed.

#### 16226       **beautify, bf**

16227 XSI      [Default *unset*]

16228       If **beautify** is set, all non-printable characters, other than <tab>, <newline>, and <form-feed>  
16229       characters, shall be discarded from text read in from files.

#### 16230       **directory, dir**

16231 MAN      [Default *implementation-dependent*]

16232       The value of this option specifies the directory in which the editor buffer is to be placed. If this  
16233       directory is not writable by the user, the editor shall quit.

|       |                         |                                                                                                                                                        |
|-------|-------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| 16234 | <b>edcompatible, ed</b> |                                                                                                                                                        |
| 16235 | MAN                     | [Default <i>unset</i> ]                                                                                                                                |
| 16236 |                         | Causes the presence of <b>g</b> and <b>c</b> suffixes on substitute commands to be remembered, and toggled                                             |
| 16237 |                         | by repeating the suffixes.                                                                                                                             |
| 16238 | <b>errorbells, eb</b>   |                                                                                                                                                        |
| 16239 |                         | [Default <i>unset</i> ]                                                                                                                                |
| 16240 |                         | If the editor is in <b>ex</b> mode, and the terminal does not support a standout mode (such as inverse                                                 |
| 16241 |                         | video), and <b>errorbells</b> is set, error messages shall be preceded by alerting the terminal.                                                       |
| 16242 | <b>exrc</b>             |                                                                                                                                                        |
| 16243 |                         | [Default <i>unset</i> ]                                                                                                                                |
| 16244 |                         | If <b>exrc</b> is set, <i>ex</i> shall access any <b>.exrc</b> file in the current directory, as described in <b>Initialization in</b>                 |
| 16245 |                         | <b>ex and vi</b> on page 392. If <b>exrc</b> is not set, <i>ex</i> shall ignore any <b>.exrc</b> file in the current directory                         |
| 16246 |                         | during initialization, unless the current directory is that named by the <i>HOME</i> environment                                                       |
| 16247 |                         | variable.                                                                                                                                              |
| 16248 | <b>ignorecase, ic</b>   |                                                                                                                                                        |
| 16249 |                         | [Default <i>unset</i> ]                                                                                                                                |
| 16250 |                         | If <b>ignorecase</b> is set, characters that have uppercase and lowercase representations shall have                                                   |
| 16251 |                         | those representations considered as equivalent for purposes of regular expression comparison.                                                          |
| 16252 |                         | The <b>ignorecase</b> edit option shall affect all remembered regular expressions; for example,                                                        |
| 16253 |                         | unsetting the <b>ignorecase</b> edit option shall cause a subsequent <i>vi n</i> command to search for the                                             |
| 16254 |                         | last basic regular expression in a case-sensitive fashion.                                                                                             |
| 16255 | <b>lisp</b>             |                                                                                                                                                        |
| 16256 | MAN                     | [Default <i>unset</i> ]                                                                                                                                |
| 16257 |                         | <b>autoindent</b> mode and the <b>(</b> , <b>)</b> , <b>{</b> , <b>}</b> , <b>[[</b> , and <b>]]</b> commands in visual mode are suitably modified for |
| 16258 |                         | <b>lisp</b> code.                                                                                                                                      |
| 16259 | <b>list</b>             |                                                                                                                                                        |
| 16260 |                         | [Default <i>unset</i> ]                                                                                                                                |
| 16261 |                         | If <b>list</b> is set, edit buffer lines written while in <i>ex</i> command mode shall be written as specified for                                     |
| 16262 |                         | the <b>print</b> command with the <b>l</b> flag specified. In open or visual mode, each edit buffer line shall                                         |
| 16263 |                         | be displayed as specified for the <i>ex print</i> command with the <b>l</b> flag specified. In open or visual                                          |
| 16264 |                         | text input mode, when the cursor does not rest on any character in the line, it shall rest on the                                                      |
| 16265 |                         | ' <b>\$</b> ' marking the end of the line.                                                                                                             |

|       |                                                                                                                   |  |
|-------|-------------------------------------------------------------------------------------------------------------------|--|
| 16266 | <b>magic</b>                                                                                                      |  |
| 16267 | [Default <i>set</i> ]                                                                                             |  |
| 16268 | If <b>magic</b> is set, modify the interpretation of characters in regular expressions and substitution           |  |
| 16269 | replacement strings (see <b>Regular Expressions in ex</b> on page 424 and <b>Replacement Strings in ex</b>        |  |
| 16270 | on page 425).                                                                                                     |  |
| 16271 | <b>mesg</b>                                                                                                       |  |
| 16272 | [Default <i>set</i> ]                                                                                             |  |
| 16273 | If <b>mesg</b> is set, the permission for others to use the <b>write</b> or <b>talk</b> commands to write to the  |  |
| 16274 | terminal shall be turned on while in open or visual mode. The shell-level command <b>mesg n</b> shall             |  |
| 16275 | take precedence over any setting of the <b>ex mesg</b> option; that is, if <b>mesg y</b> was issued before the    |  |
| 16276 | editor started (or in a shell escape), such as:                                                                   |  |
| 16277 | <code>:!mesg y</code>                                                                                             |  |
| 16278 | the <b>mesg</b> option in <b>ex</b> shall suppress incoming messages, but the <b>mesg</b> option shall not enable |  |
| 16279 | incoming messages if <b>mesg n</b> was issued.                                                                    |  |
| 16280 | <b>number, nu</b>                                                                                                 |  |
| 16281 | [Default <i>unset</i> ]                                                                                           |  |
| 16282 | If <b>number</b> is set, edit buffer lines written while in <b>ex</b> command mode shall be written with line     |  |
| 16283 | numbers, in the format specified by the <b>print</b> command with the # flag specified. In <b>ex</b> text input   |  |
| 16284 | mode, each line shall be preceded by the line number it will have in the file.                                    |  |
| 16285 | In open or visual mode, each edit buffer line shall be displayed with a preceding line number, in                 |  |
| 16286 | the format specified by the <b>ex print</b> command with the # flag specified. This line number shall             |  |
| 16287 | not be considered part of the line for the purposes of evaluating the current column; that is,                    |  |
| 16288 | column position 1 shall be the first column position after the format specified by the <b>print</b>               |  |
| 16289 | command.                                                                                                          |  |
| 16290 | <b>paragraphs, para</b>                                                                                           |  |
| 16291 | [Default in the POSIX locale <code>IPLPPPQPP LIpplpipbp</code> ]                                                  |  |
| 16292 | The <b>paragraphs</b> edit option shall define additional paragraph boundaries for the open and visual            |  |
| 16293 | mode commands. The <b>paragraphs</b> edit option can be set to a character string consisting of zero              |  |
| 16294 | or more character pairs. It shall be an error to set it to an odd number of characters.                           |  |
| 16295 | <b>prompt</b>                                                                                                     |  |
| 16296 | [Default <i>set</i> ]                                                                                             |  |
| 16297 | If <b>prompt</b> is set, <b>ex</b> command mode input shall be prompted for with a colon (':'); when unset,       |  |
| 16298 | no prompt shall be written.                                                                                       |  |



16299      **readonly**16300      [Default *see text*]

16301      If **readonly** edit option is set, read-only mode shall be enabled (see **Write** on page 419). The  
 16302      **readonly** edit option shall be initialized to set if either of the following conditions are true:

- 16303      • The command-line option **-R** was specified.
- 16304      • Performing actions equivalent to the *access()* function called with the following arguments  
 16305      indicates that the file lacks write permission:
  - 16306          1. The current path name is used as the *path* argument.
  - 16307          2. The constant **W\_OK** is used as the *amode* argument.

16308      The **readonly** edit option may be initialized to set for other, implementation-dependent reasons.  
 16309      The **readonly** edit option shall not be initialized to unset based on any special privileges of the  
 16310      user or process. The **readonly** edit option shall be reinitialized each time that the contents of the  
 16311      edit buffer are replaced (for example, by an **edit** or **next** command) unless the user has explicitly  
 16312      set it, in which case it shall remain set until the user explicitly unsets it. Once unset, it shall again  
 16313      be reinitialized each time that the contents of the edit buffer are replaced.

16314      **redraw**16315      MAN [Default *unset*]

16316      The editor simulates an intelligent terminal on a dumb terminal. (Since this is likely to require a  
 16317      large amount of output to the terminal, it is useful only at high transmission speeds.)

16318      **remap**16319      [Default *set*]

16320      If **remap** is set, map translation shall allow for maps defined in terms of other maps; translation  
 16321      shall continue until a final product is obtained. If unset, only a one-step translation shall be done.

16322      **report**

16323      [Default 5]

16324      The value of this **report** edit option specifies what number of lines being added, copied, deleted,  
 16325      or modified in the edit buffer will cause an informational message to be written to the user. The  
 16326      following conditions shall cause an informational message. The message shall contain the  
 16327      number of lines added, copied, deleted, or modified, but is otherwise unspecified.

- 16328      • An *ex* or *vi* editor command, other than **open**, **undo**, or **visual**, that modifies at least the value  
 16329      of the **report** edit option number of lines, and which is not part of an *ex* **global** or **v**  
 16330      command, or *ex* or *vi* buffer execution, shall cause an informational message to be written.
- 16331      • An *ex* **yank** or *vi* **y** or **Y** command, that copies at least the value of the **report** edit option plus  
 16332      1 number of lines, and which is not part of an *ex* **global** or **v** command, or *ex* or *vi* buffer  
 16333      execution, shall cause an informational message to be written.
- 16334      • An *ex* **global**, **v**, **open**, **undo**, or **visual** command or *ex* or *vi* buffer execution, that adds or  
 16335      deletes a total of at least the value of the **report** edit option number of lines, and which is not  
 16336      part of an *ex* **global** or **v** command, or *ex* or *vi* buffer execution, shall cause an informational  
 16337      message to be written. (For example, if 3 lines were added and 8 lines deleted during an *ex*  
 16338      **visual** command, 5 would be the number compared against the **report** edit option after the  
 16339      command completed.

|       |                                                                                                                         |  |
|-------|-------------------------------------------------------------------------------------------------------------------------|--|
| 16340 | <b>scroll, scr</b>                                                                                                      |  |
| 16341 | [Default (number of lines in the display -1)/2]                                                                         |  |
| 16342 | The value of the <b>scroll</b> edit option shall determine the number of lines scrolled by by the <i>ex</i>             |  |
| 16343 | <control>-D and <b>z</b> commands. For the <i>vi</i> <control>-D and <control>-U commands, it shall be the              |  |
| 16344 | initial number of lines to scroll when no previous <control>-D or <control>-U command has                               |  |
| 16345 | been executed.                                                                                                          |  |
| 16346 | <b>sections</b>                                                                                                         |  |
| 16347 | [Default in the POSIX locale <code>NHSHH HUnhsh</code> ]                                                                |  |
| 16348 | The <b>sections</b> edit option shall define additional section boundaries for the open and visual mode                 |  |
| 16349 | commands. The <b>sections</b> edit option can be set to a character string consisting of zero or more                   |  |
| 16350 | character pairs; it shall be an error to set it to an odd number of characters.                                         |  |
| 16351 | <b>shell, sh</b>                                                                                                        |  |
| 16352 | [Default from the environment variable <i>SHELL</i> ]                                                                   |  |
| 16353 | The value of this option shall be a string. The default shall be taken from the <i>SHELL</i>                            |  |
| 16354 | environment variable. If the <i>SHELL</i> environment variable is null or empty, the <i>sh</i> (see <i>sh</i> ) utility |  |
| 16355 | shall be the default.                                                                                                   |  |
| 16356 | <b>shiftwidth, sw</b>                                                                                                   |  |
| 16357 | [Default 8]                                                                                                             |  |
| 16358 | The value of this option shall give the width in columns of an indentation level used during                            |  |
| 16359 | autoindentation and by the shift commands (< and >).                                                                    |  |
| 16360 | <b>showmatch, sm</b>                                                                                                    |  |
| 16361 | [Default <i>unset</i> ]                                                                                                 |  |
| 16362 | The functionality described for the <b>showmatch</b> edit option need not be supported on block-                        |  |
| 16363 | mode terminals or terminals with insufficient capabilities.                                                             |  |
| 16364 | If <b>showmatch</b> is set, in open or visual mode, when a ' ) ' or ' } ' is typed, if the matching ' ( ' or            |  |
| 16365 | ' { ' is currently visible on the display, the matching ' ( ' or ' { ' shall be flagged moving the                      |  |
| 16366 | cursor to its location for an unspecified amount of time.                                                               |  |
| 16367 | <b>showmode</b>                                                                                                         |  |
| 16368 | [Default <i>unset</i> ]                                                                                                 |  |
| 16369 | If <b>showmode</b> is set, in open or visual mode, the current mode that the editor is in shall be                      |  |
| 16370 | displayed on the last line of the display. Command mode and text input mode shall be                                    |  |
| 16371 | differentiated; other unspecified modes and implementation-dependent information may be                                 |  |
| 16372 | displayed.                                                                                                              |  |

|       |                                                                                                           |  |
|-------|-----------------------------------------------------------------------------------------------------------|--|
| 16373 | <b>slowopen</b>                                                                                           |  |
| 16374 | [Default <i>unset</i> ]                                                                                   |  |
| 16375 | If <b>slowopen</b> is set during open and visual text input modes, the editor shall not update portions   |  |
| 16376 | of the display other than those screen columns that display the characters entered by the user            |  |
| 16377 | (see <b>Input Mode Commands in vi</b> on page 1064).                                                      |  |
| 16378 | <b>tabstop, ts</b>                                                                                        |  |
| 16379 | [Default 8]                                                                                               |  |
| 16380 | The value of this edit option shall specify the column boundary used by a <tab> character in the          |  |
| 16381 | display (see <b>autoprint, ap</b> on page 426 and <b>Input Mode Commands in vi</b> on page 1064).         |  |
| 16382 | <b>taglength, tl</b>                                                                                      |  |
| 16383 | [Default zero]                                                                                            |  |
| 16384 | The value of this edit option shall specify the maximum number of characters that are                     |  |
| 16385 | considered significant in the user-specified tag name and in the tag name from the tags file. If the      |  |
| 16386 | value is zero, all characters in both tag names shall be significant.                                     |  |
| 16387 | <b>tags</b>                                                                                               |  |
| 16388 | [Default <i>see text</i> ]                                                                                |  |
| 16389 | The value of this edit option shall be a string of <blank> character-delimited path names of files        |  |
| 16390 | used by the <b>tag</b> command. The default value is unspecified.                                         |  |
| 16391 | <b>term</b>                                                                                               |  |
| 16392 | [Default from the environment variable <i>TERM</i> ]                                                      |  |
| 16393 | The value of this edit option shall be a string. The default shall be taken from the <i>TERM</i> variable |  |
| 16394 | in the environment. If the <i>TERM</i> environment variable is empty or null, the default is              |  |
| 16395 | unspecified. The editor shall use the value of this edit option to determine the type of the display      |  |
| 16396 | device.                                                                                                   |  |
| 16397 | The results are unspecified if the user changes the value of the term edit option after editor            |  |
| 16398 | initialization.                                                                                           |  |
| 16399 | <b>terse</b>                                                                                              |  |
| 16400 | [Default <i>unset</i> ]                                                                                   |  |
| 16401 | If <b>terse</b> is set, error messages may be less verbose. However, except for this caveat, error        |  |
| 16402 | messages are unspecified. Furthermore, not all error messages need change for different settings          |  |
| 16403 | of this option.                                                                                           |  |
| 16404 | <b>warn</b>                                                                                               |  |
| 16405 | [Default <i>set</i> ]                                                                                     |  |
| 16406 | If <b>warn</b> is set, and the contents of the edit buffer have been modified since they were last        |  |
| 16407 | completely written, the editor shall write a warning message before certain ! commands (see               |  |
| 16408 | <b>Escape</b> on page 422).                                                                               |  |

|       |                                                                                                               |  |
|-------|---------------------------------------------------------------------------------------------------------------|--|
| 16409 | <b>window</b>                                                                                                 |  |
| 16410 | [Default <i>see text</i> ]                                                                                    |  |
| 16411 | A value used in open and visual mode, by the <control>-B and <control>-F commands, and, in                    |  |
| 16412 | visual mode, to specify the number of lines displayed when the screen is repainted.                           |  |
| 16413 | If the <b>-w</b> command-line option is not specified, the default value shall be set to the value of the     |  |
| 16414 | <i>LINES</i> environment variable. If the <i>LINES</i> environment variable is empty or null, the default     |  |
| 16415 | shall be the number of lines in the display minus 1.                                                          |  |
| 16416 | Setting the <b>window</b> edit option to zero or to a value greater than the number of lines in the           |  |
| 16417 | display minus 1 (either explicitly or based on the <b>-w</b> option or the <i>LINES</i> environment variable) |  |
| 16418 | shall cause the <b>window</b> edit option to be set to the number of lines in the display minus 1.            |  |
| 16419 | The baud rate of the terminal line may change the default in an implementation-dependent                      |  |
| 16420 | manner.                                                                                                       |  |
| 16421 | <b>wrapmargin, wm</b>                                                                                         |  |
| 16422 | [Default 0]                                                                                                   |  |
| 16423 | If the value of this edit option is zero, it shall have no effect.                                            |  |
| 16424 | If not in the POSIX locale, the effect of this edit option is implementation-dependent.                       |  |
| 16425 | Otherwise, it shall specify a number of columns from the ending margin of the terminal.                       |  |
| 16426 | During open and visual text input modes, for each character for which any part of the character               |  |
| 16427 | is displayed in a column that is less than <b>wrapmargin</b> columns from the ending margin of the            |  |
| 16428 | screen, the editor shall behave as follows:                                                                   |  |
| 16429 | 1. If the character triggering this event is a <blank> character, it, and all immediately                     |  |
| 16430 | preceding <blank> characters on the current line entered during the execution of the                          |  |
| 16431 | current text input command, shall be discarded, and the editor shall behave as if the user                    |  |
| 16432 | had entered a single <newline> character instead. In addition, if the next user-entered                       |  |
| 16433 | character is a <space> character, it shall be discarded as well.                                              |  |
| 16434 | 2. Otherwise, if there are one or more <blank> characters on the current line immediately                     |  |
| 16435 | preceding the last group of inserted non-<blank> characters which was entered during the                      |  |
| 16436 | execution of the current text input command, the <blank> characters shall be replaced as if                   |  |
| 16437 | the user had entered a single <newline> character instead.                                                    |  |
| 16438 | If the <b>autoindent</b> edit option is set, and the events described in 1. or 2. are performed, any          |  |
| 16439 | <blank> characters at or after the cursor in the current line shall be discarded.                             |  |
| 16440 | The ending margin shall be determined by the system or overridden by the user, as described for               |  |
| 16441 | <i>COLUMNS</i> in the ENVIRONMENT VARIABLES section and the System Interface Definitions                      |  |
| 16442 | volume of IEEE Std. 1003.1-200x, Chapter 8, Environment Variables.                                            |  |
| 16443 | <b>wrapscan, ws</b>                                                                                           |  |
| 16444 | [Default <i>set</i> ]                                                                                         |  |
| 16445 | If <b>wrapscan</b> is set, searches (the <i>ex</i> / or ? addresses, or open and visual mode /, ?, N, and n   |  |
| 16446 | commands) shall wrap around the beginning or end of the edit buffer; when unset, searches                     |  |
| 16447 | shall stop at the beginning or end of the edit buffer.                                                        |  |

16448       **writeany, wa**  
 16449       [Default *unset*]

16450       If **writeany** is set, some of the checks performed when executing the **ex write** commands shall be  
 16451       inhibited, as described in editor option **autowrite**.

16452 **EXIT STATUS**  
 16453       The following exit values shall be returned:

16454       0   Successful completion.  
 16455       >0   An error occurred.

16456 **CONSEQUENCES OF ERRORS**  
 16457       When any error is encountered and the standard input is not a terminal device file, **ex** shall not  
 16458       write the file or return to command or text input mode, and shall terminate with a non-zero exit  
 16459       status.

16460       Otherwise, when an unrecoverable error is encountered, it shall be equivalent to a SIGHUP  
 16461       asynchronous event.

16462       Otherwise, when an error is encountered, the editor shall behave as specified in **Command Line**  
 16463       **Parsing in ex** on page 396.

16464 **APPLICATION USAGE**  
 16465       If a SIGSEGV signal is received while **ex** is saving a file, the file might not be successfully saved.

16466       The **next** command can accept more than one file, so usage such as:

16467       next `ls [abc]\*`

16468       is valid; it would not be valid for the **edit** or **read** commands, for example, because they expect  
 16469       only one file and unspecified results occur.

16470       Application writers should note that this utility need not be provided on systems that do not  
 16471       support the User Portability Utilities option.

16472 **EXAMPLES**  
 16473       None.

16474 **RATIONALE**  
 16475       The **ex/vi** specification is based on the historical practice found in the 4 BSD and System V  
 16476       implementations of **ex** and **vi**. A freely redistributable implementation of **ex/vi**, which is  
 16477       tracking IEEE Std. 1003.1-200x fairly closely, and demonstrates the intended changes between  
 16478       historical implementations and IEEE Std. 1003.1-200x, may be obtained from Keith Bostic  
 16479       ([bostic@cs.berkeley.edu](mailto:bostic@cs.berkeley.edu)) or by anonymous FTP from:

16480       ftp.cs.berkeley.edu:ucb/4bsd/nvi.tar.gz

16481       A *restricted editor* (both the historical *red* utility and modifications to **ex**) were considered and  
 16482       rejected for inclusion. Neither option provided the level of security that users might expect.

16483       It is recognized that **ex** visual mode and related features would be difficult, if not impossible, to  
 16484       implement satisfactorily on a block-mode terminal, or a terminal without any form of cursor  
 16485       addressing; thus, it is not a mandatory requirement that such features should work on all  
 16486       terminals. It is the intention, however, that an **ex** implementation should provide the full set of  
 16487       capabilities on all terminals capable of supporting them.

## Options

The **-c** replacement for *+command* was inspired by the **-e** option of *sed*. Historically, all such commands (see **edit** and **next** as well) were executed from the last line of the edit buffer. This meant, for example, that *+/pattern* would fail unless the **wraps** option was set. IEEE Std. 1003.1-200x requires conformance to historical practice. Historically, some implementations restricted the *ex* commands that could be listed as part of the command line arguments. For consistency, IEEE Std. 1003.1-200x does not permit these restrictions.

In historical implementations of the editor, the **-R** option (and the **readonly** edit option) only prevented overwriting of files; appending to files was still permitted, mapping loosely into the *cs*h **noclobber** variable. Some implementations, however, have not followed this semantic, and **readonly** does not permit appending either. IEEE Std. 1003.1-200x follows the latter practice, believing that it is a more obvious and intuitive meaning of **readonly**.

The **-s** option suppresses all interactive user feedback and is useful for editing scripts in batch jobs. The list of specific effects is historical practice. The terminal type “incapable of supporting open and visual modes” has historically been named “dumb”.

The **-t** option was required because the *ctags* utility appears in IEEE Std. 1003.1-200x and the option is available in all historical implementations of *ex*.

Historically, the *ex* and *vi* utilities accepted a **-x** option, which did encryption based on the algorithm found in the historical *crypt* utility. The **-x** option for encryption, and the associated *crypt* utility, were omitted because the algorithm used was not specifiable and the export control laws of some nations make it difficult to export cryptographic technology. In addition, it did not historically provide the level of security that users might expect.

## Standard Input

An end-of-file condition is not equivalent to an end-of-file character. A common end-of-file character, <control>-D, is historically an *ex* command.

There was no maximum line length in historical implementations of *ex*. Specifically, as it was parsed in chunks, the addresses had a different maximum length than the file names. Further, the maximum line buffer size was declared as {BUFSIZ}, which was different lengths on different systems. This version selected the value of {LINE\_MAX} to impose a reasonable restriction on portable usage of *ex* and to aid test suite writers in their development of realistic tests that exercise this limit.

## Input Files

It was an explicit decision by the standard developers that a <newline> character be added to any file lacking one. It was believed that this feature of *ex* and *vi* was relied on by users in order to make text files lacking a trailing <newline> more portable. It is recognized that this will require a user-specified option or extension for implementations that permit *ex* and *vi* to edit files of type other than text if such files are not otherwise identified by the system. It was agreed that the ability to edit files of arbitrary type can be useful, but it was not considered necessary to mandate that an *ex* or *vi* implementation be required to handle files other than text files.

The paragraph in the INPUT FILES section, “By default, ...”, is intended to close a long-standing security problem in *ex* and *vi*, that of the “modeline” or “modelines” edit option. This feature allows any line in the first or last five lines of the file containing the strings “*ex*: ” or “*vi*: ” (and, apparently, “*ei*: ” or “*vx*: ”) to be a line containing editor commands, and *ex* interprets all the text up to the next ‘:’ or <newline> as a command. Consider the consequences, for example, of an unsuspecting user using *ex* or *vi* as the editor when replying to a mail message in which a line such as:

16534 `ex:! rm -rf :`

16535 appeared in the signature lines. The standard developers believed strongly that an editor should  
 16536 not by default interpret any lines of a file. Vendors are strongly urged to delete this feature from  
 16537 their implementations of *ex* and *vi*.

## 16538 **Asynchronous Events**

16539 The intention of the phrase “complete write” is that the entire edit buffer be written to stable  
 16540 storage. The note regarding temporary files is intended for implementations that use temporary  
 16541 files to back edit buffers unnamed by the user.

16542 Historically, SIGQUIT was ignored by *ex*, but was the equivalent of the **Q** command in visual  
 16543 mode; that is, it exited visual mode and entered *ex* mode. IEEE Std. 1003.1-200x permits, but does  
 16544 not require, this behavior. Historically, SIGINT was often used by *vi* users to terminate text  
 16545 input mode (<control>-C is often easier to enter than <ESC>). Some implementations of *vi*  
 16546 alerted the terminal on this event, and some did not. IEEE Std. 1003.1-200x requires that SIGINT  
 16547 behave identically to <ESC>, and that the terminal not be alerted.

16548 Historically, suspending the *ex* editor during text input mode was similar to SIGINT, as  
 16549 completed lines were retained, but any partial line discarded, and the editor returned to  
 16550 command mode. IEEE Std. 1003.1-200x is silent on this issue; implementations are encouraged to  
 16551 follow historical practice, where possible.

16552 Historically, the *vi* editor did not treat SIGTSTP as an asynchronous event, and it was therefore  
 16553 impossible to suspend the editor in visual text input mode. There are two major reasons for this.  
 16554 The first is that SIGTSTP is a broadcast signal on UNIX systems, and the chain of events where  
 16555 the shell *execs* an application that then *execs vi* usually caused confusion for the terminal state if  
 16556 SIGTSTP was delivered to the process group in the default manner. The second was that most  
 16557 implementations of the UNIX *curses* package are not reentrant, and the receipt of SIGTSTP at the  
 16558 wrong time will cause them to crash. IEEE Std. 1003.1-200x is silent on this issue;  
 16559 implementations are encouraged to treat suspension as an asynchronous event if possible.

16560 Historically, modifications to the edit buffer made before SIGINT interrupted an operation were  
 16561 retained; that is, anywhere from zero to all of the lines to be modified might have been modified  
 16562 by the time the SIGINT arrived. These changes were not discarded by the arrival of SIGINT.  
 16563 IEEE Std. 1003.1-200x permits this behavior, noting that the *undo* command is required to be able  
 16564 to undo these partially completed commands.

16565 The action taken for signals other than SIGINT, SIGCONT, SIGHUP, and SIGTERM is  
 16566 unspecified because some implementations attempt to save the edit buffer in a useful state when  
 16567 other signals are received.

## 16568 **Standard Error**

16569 For *ex/vi*, diagnostic messages are those messages reported as a result of a failed attempt to  
 16570 invoke *ex* or *vi*, such as invalid options or insufficient resources, or an abnormal termination  
 16571 condition. Diagnostic messages should not be confused with the error messages generated by  
 16572 inappropriate or illegal user commands.

## Initialization in ex and vi

If an **ex** command (other than **cd**, **chdir**, or **source**) has a file name argument, one or both of the alternate and current path names will be set. Informally, they are set as follows:

1. If the **ex** command is one that replaces the contents of the edit buffer, and it succeeds, the current path name will be set to the file name argument (the first file name argument in the case of the **next** command) and the alternate path name will be set to the previous current path name, if there was one.
2. In the case of the file read/write forms of the **read** and **write** commands, if there is no current path name, the current path name will be set to the file name argument.
3. Otherwise, the alternate path name will be set to the file name argument.

For example, **:edit foo** and **:recover foo**, when successful, set the current path name, and, if there was a previous current path name, the alternate path name. The commands **:write**, **!command**, and **:edit** set neither the current or alternate path names. If the **:edit foo** command were to fail for some reason, the alternate path name would be set. The **read** and **write** commands set the alternate path name to their *file* argument, unless the current path name is not set, in which case they set the current path name to their *file* arguments. The alternate path name was not historically set by the **:source** command.  $\zeta$ A requires conformance to historical practice. Implementations adding commands that take file names as arguments are encouraged to set the alternate path name as described here.

Historically, **ex** and **vi** read the **.exrc** file in the *\$HOME* directory twice, if the editor was executed in the *\$HOME* directory. IEEE Std. 1003.1-200x prohibits this behavior.

Historically, the 4 BSD **ex** and **vi** read the *\$HOME* and local **.exrc** files if they were owned by the real ID of the user, or the **sourceany** option was set, regardless of other considerations. This was a security problem because it is possible to put normal UNIX system commands inside a **.exrc** file. IEEE Std. 1003.1-200x does not specify the **sourceany** option, and historical implementations are encouraged to delete it.

The **.exrc** files must be owned by the real ID of the user, and not writeable by anyone other than the owner. The appropriate privileges exception is intended to permit users to acquire special privileges, but continue to use the **.exrc** files in their home directories.

System V Release 3.2 and later **vi** implementations added the option **[no]exrc**. The behavior is that local **.exrc** files are read-only if the **exrc** option is set. The default for the **exrc** option was off, so by default, local **.exrc** files were not read. The problem this was intended to solve was that System V permitted users to give away files, so there is no possible ownership or writeability test to ensure that the file is safe. This is still a security problem on systems where users can give away files, but there is nothing additional that IEEE Std. 1003.1-200x can do. The implementation-dependent exception is intended to permit groups to have local **.exrc** files that are shared by users, by creating pseudo-users to own the shared files.

IEEE Std. 1003.1-200x does not mention system-wide **ex** and **vi** start-up files. While they exist in several implementations of **ex** and **vi**, they are not present in any implementations considered historical practice by IEEE Std. 1003.1-200x. Implementations that have such files should use them only if they are owned by the real user ID or an appropriate user (for example, root on UNIX systems) and if they are not writeable by any user other than their owner. System-wide start-up files should be read before the *EXINIT* variable, *\$HOME/.exrc* or local **.exrc** files are evaluated.

Historically, any **ex** command could be entered in the *EXINIT* variable or the **.exrc** file, although ones requiring that the edit buffer already contain lines of text generally caused historical implementations of the editor to drop core. IEEE Std. 1003.1-200x requires that any **ex** command



16620 be permitted in the *EXINIT* variable and *.exrc* files, for simplicity of specification and  
 16621 consistency, although many of them will obviously fail under many circumstances.

16622 The initialization of the contents of the edit buffer uses the phrase “the effect shall be” with  
 16623 regard to various *ex* commands. The intent of this phrase is that edit buffer contents loaded  
 16624 during the initialization phase not be lost; that is, loading the edit buffer should fail if the *.exrc*  
 16625 file read in the contents of a file and did not subsequently write the edit buffer. An additional  
 16626 intent of this phrase is to specify that the initial current line and column is set as specified for the  
 16627 individual *ex* commands.

16628 Historically, the *-t* option behaved as if the tag search were a *+command*; that is, it was executed  
 16629 from the last line of the file specified by the tag. This resulted in the search failing if the pattern  
 16630 was a forward search pattern and the *wraps* edit option was not set. IEEE Std. 1003.1-200x  
 16631 does not permit this behavior, requiring that the search for the tag pattern be performed on the  
 16632 entire file, and, if not found, that the current line be set to a more reasonable location in the file.

16633 Historically, the empty edit buffer presented for editing when a file was not specified by the user  
 16634 was unnamed. This is permitted by IEEE Std. 1003.1-200x; however, implementations are  
 16635 encouraged to provide users a temporary file name for this buffer because it permits them the  
 16636 use of *ex* commands that use the current path name during temporary edit sessions.

16637 Historically, the file specified using the *-t* option was not part of the current argument list. This  
 16638 practice is permitted by IEEE Std. 1003.1-200x; however, implementations are encouraged to  
 16639 include its name in the current argument list for consistency.

16640 Historically, the *-c* command was generally not executed until a file that already exists was  
 16641 edited. IEEE Std. 1003.1-200x requires conformance to this historical practice. Commands that  
 16642 could cause the *-c* command to be executed include the *ex* commands **edit**, **next**, **recover**,  
 16643 **rewind**, and **tag**, and the *vi* commands *<control>^* and *<control>-]*. Historically, reading a file  
 16644 into an edit buffer did not cause the *-c* command to be executed (even though it might set the  
 16645 current path name) with the exception that it did cause the *-c* command to be executed if: the  
 16646 editor was in *ex* mode, the edit buffer had no current path name, the edit buffer was empty, and  
 16647 no read commands had yet been attempted. For consistency and simplicity of specification,  
 16648 IEEE Std. 1003.1-200x does not permit this behavior.

16649 Historically, the *-r* option was the same as a normal edit session if there was no recovery  
 16650 information available for the file. This allowed users to enter:

16651 `vi -r *.c`

16652 and recover whatever files were recoverable. In some implementations, recovery was attempted  
 16653 only on the first file named, and the file was not entered into the argument list; in others,  
 16654 recovery was attempted for each file named. In addition, some historical implementations  
 16655 ignored *-r* if *-t* was specified or did not support command line *file* arguments with the *-t* option.  
 16656 For consistency and simplicity of specification, IEEE Std. 1003.1-200x disallows these special  
 16657 cases, and requires that recovery be attempted the first time each file is edited.

16658 Historically, *vi* initialized the ‘ and ’ marks, but *ex* did not. This meant that if the first command  
 16659 in *ex* mode was **visual** or if an *ex* command was executed first (for example, *vi +10 file*), *vi* was  
 16660 entered without the marks being initialized. Because the standard developers believed the marks  
 16661 to be generally useful, and for consistency and simplicity of specification, IEEE Std. 1003.1-200x  
 16662 requires that they always be initialized if in open or visual mode, or if in *ex* mode and the edit  
 16663 buffer is not empty. Not initializing it in *ex* mode if the edit buffer is empty is historical practice;  
 16664 however, it has always been possible to set (and use) marks in empty edit buffers in open and  
 16665 visual mode edit sessions.

## Addressing

Historically, *ex* and *vi* accepted the additional addressing forms '`\`' and '`?`'. They were equivalent to '`//`' and '`??`', respectively. They are not required by IEEE Std. 1003.1-200x, mostly because nobody can remember whether they ever did anything different historically.

Historically, *ex* and *vi* permitted an address of zero for several commands, and permitted the % address in empty files for others. For consistency, IEEE Std. 1003.1-200x requires support for the former in the few commands where it makes sense, and disallows it otherwise. In addition, because IEEE Std. 1003.1-200x requires that % be logically equivalent to '`1,$`', it is also supported where it makes sense and disallowed otherwise.

Historically, the % address could not be followed by further addresses. For consistency and simplicity of specification, IEEE Std. 1003.1-200x requires that additional addresses be supported.

All of the following are valid *addresses*:

- +++ Three lines after the current line.
- /re/- One line before the next occurrence of *re*.
- 2 Two lines before the current line.
- 3 ——— 2 Line one (note intermediate negative address).
- 1 2 3 Line six.

Any number of addresses can be provided to commands taking addresses; for example, '`1,2,3,4,5p`' prints lines 4 and 5, because two is the greatest valid number of addresses accepted by the **print** command. This, in combination with the semicolon delimiter, permits users to create commands based on ordered patterns in the file. For example, the command **3/foo/+2print** will display the first line after line 3 that contains the pattern *foo*, plus the next two lines. Note that the address **3**; must be evaluated before being discarded because the search origin for the **/foo/** command depends on this.

Historically, values could be added to addresses by including them after one or more <blank> characters; for example, **3 - 5p** wrote the seventh line of the file, and **/foo/ 5** was the same as **/foo/+5**. However, only absolute values could be added; for example, **5 /foo/** was an error. IEEE Std. 1003.1-200x requires conformance to historical practice. Address offsets are separately specified from addresses because they could historically be provided to visual mode search commands.

Historically, any missing addresses defaulted to the current line. This was true for leading and trailing comma-delimited addresses, and for trailing semicolon-delimited addresses. For consistency, IEEE Std. 1003.1-200x requires it for leading semicolon addresses as well.

Historically, *ex* and *vi* accepted the '`^`' character as both an address and as a flag offset for commands. In both cases it was identical to the '`-`' character. IEEE Std. 1003.1-200x does not require or prohibit this behavior.

Historically, the enhancements to basic regular expressions could be used in addressing; for example, '`~`', '`\<`', and '`\>`'. IEEE Std. 1003.1-200x requires conformance to historical practice; that is, that regular expression usage be consistent, and that regular expression enhancements be supported wherever regular expressions are used.

## Command Line Parsing in ex

Historical **ex** command parsing was even more complex than that described here. IEEE Std. 1003.1-200x requires the subset of the command parsing that the standard developers believed was documented and that users could reasonably be expected to use in a portable fashion, and that was historically consistent between implementations. (The discarded functionality is obscure, at best.) Historical implementations will require changes in order to comply with IEEE Std. 1003.1-200x; however, users are not expected to notice any of these changes. Most of the complexity in **ex** parsing is to handle three special termination cases:

1. The **!**, **global**, **v**, and the filter versions of the **read** and **write** commands are delimited by <newline> characters (they can contain vertical-line characters that are usually shell pipes).
2. The **ex**, **edit**, **next**, and **visual** in open and visual mode commands all take **ex** commands, optionally containing vertical-line characters, as their first arguments.
3. The **s** command takes a regular expression as its first argument, and uses the delimiting characters to delimit the command.

Historically, vertical-line characters in the *+command* argument of the **ex**, **edit**, **next**, **vi**, and **visual** commands, and in the *pattern* and *replacement* parts of the **s** command, did not delimit the command, and in the filter cases for **read** and **write**, and the **!**, **global**, and **v** commands, they did not delimit the command at all. For example, the following commands are all valid:

```
:edit +25 | s/abc/ABC/ file.c
:s/ | /PIPE/
:read !spell % | columnate
:global/pattern/p | l
:s/a/b/ | s/c/d | set
```

Historically, empty or <<blank>> filled lines in **.exrc** files and **sourced** files (as well as **EXINIT** variables and **ex** command scripts) were treated as default commands; that is, **print** commands. IEEE Std. 1003.1-200x specifically requires that they be ignored when encountered in **.exrc** and **sourced** files to eliminate a common source of new user error.

Historically, **ex** commands with multiple adjacent (or <blank>-separated) vertical lines were handled oddly when executed from **ex** mode. For example, the command `||| <carriage-return>`, when the cursor was on line 1, displayed lines 2, 3, and 5 of the file. In addition, the command `|` would only display the line after the next line, instead of the next two lines. The former worked more logically when executed from **vi** mode, and displayed lines 2, 3, and 4. IEEE Std. 1003.1-200x requires the **vi** behavior; that is, a single default command and line number increment for each command separator, and trailing <newline> characters after vertical-line separators are discarded.

Historically, **ex** permitted a single extra colon as a leading command character; for example, **:g/pattern/p** was a valid command. IEEE Std. 1003.1-200x generalizes this to require that any number of leading colon characters be stripped.

Historically, any prefix of the **delete** command could be followed without intervening <blank> characters by a flag character because in the command **d p**, **p** is interpreted as the buffer **p**. IEEE Std. 1003.1-200x requires conformance to historical practice.

Historically, the **k** command could be followed by the mark name without intervening <blank> characters. IEEE Std. 1003.1-200x requires conformance to historical practice.

Historically, the **s** command could be immediately followed by flag and option characters; for example, **s/e/E/|s|sgc3p** was a valid command. However, flag characters could not stand alone; for example, the commands **sp** and **s l** would fail, while the command **sgp** and **s gl** would

16753 succeed. (Obviously, the ' #' flag character was used as a delimiter character if it followed the  
 16754 command.) Another issue was that option characters had to precede flag characters even when  
 16755 the command was fully specified; for example, the command **s/e/E/pg** would fail, while the  
 16756 command **s/e/E/gp** would succeed. IEEE Std. 1003.1-200x requires conformance to historical  
 16757 practice.

16758 Historically, the first command name that had a prefix matching the input from the user was the  
 16759 executed command; for example, **ve**, **ver**, and **vers** all executed the **version** command.  
 16760 Commands were in a specific order, however, so that **a** matched **append**, not **abbreviate**.  
 16761 IEEE Std. 1003.1-200x requires conformance to historical practice. The restriction on command  
 16762 search order for implementations with extensions is to avoid the addition of commands such  
 16763 that the historical prefixes would fail to work portably.

16764 Historical implementations of *ex* and *vi* did not correctly handle multiple *ex* commands,  
 16765 separated by vertical-line characters, that entered or exited visual mode or the editor. Because  
 16766 implementations of *vi* exist that do not exhibit this failure mode, IEEE Std. 1003.1-200x does not  
 16767 permit it.

16768 The requirement that alphabetic command names consist of all following alphabetic characters  
 16769 up to the next non-alphabetic character means that alphabetic command names must be  
 16770 separated from their arguments by one or more non-alphabetic characters, normally a <blank>  
 16771 or '!' character, except as specified for the exceptions, the **delete**, **k**, and **s** commands.

16772 Historically, the repeated execution of the *ex* default **print** commands (<control>-D, *eof*,  
 16773 <newline>, <carriage-return>) erased any prompting character and displayed the next lines  
 16774 without scrolling the terminal; that is, immediately below any previously displayed lines. This  
 16775 provided a cleaner presentation of the lines in the file for the user. IEEE Std. 1003.1-200x does not  
 16776 require this behavior because it may be impossible in some situations; however,  
 16777 implementations are strongly encouraged to provide this semantic if possible.

16778 Historically, it was possible to change files in the middle of a command, and have the rest of the  
 16779 command executed in the new file; for example:

16780 `:edit +25 file.c | s/abc/ABC/ | 1`

16781 was a valid command, and the substitution was attempted in the newly edited file.  
 16782 IEEE Std. 1003.1-200x requires conformance to historical practice. The following commands are  
 16783 examples that exercise the *ex* parser:

16784 `echo 'foo | bar' > file1; echo 'foo/bar' > file2;`

16785 `vi`

16786 `:edit +1 | s/|/PIPE/ | w file1 | e file2 | 1 | s/\\//SLASH/ | wq`

16787 Historically, there was no protection in editor implementations to avoid **ex global**, **v**, **@**, or **\***  
 16788 commands changing edit buffers during execution of their associated commands. Because this  
 16789 would almost invariably result in catastrophic failure of the editor, and implementations exist  
 16790 that do exhibit these problems, IEEE Std. 1003.1-200x requires that changing the edit buffer  
 16791 during a **global** or **v** command, or during a **@** or **\*** command for which there will be more than a  
 16792 single execution, be an error. Implementations supporting multiple edit buffers simultaneously  
 16793 are strongly encouraged to apply the same semantics to switching between buffers as well.

16794 The *ex* command quoting required by IEEE Std. 1003.1-200x is a superset of the quoting in  
 16795 historical implementations of the editor. For example, it was not historically possible to escape a  
 16796 <blank> character in a file name; for example, `:edit foo\\\ bar` would report that too many file  
 16797 names had been entered for the edit command, and there was no method of escaping a <blank>  
 16798 in the first argument of an **edit**, **ex**, **next**, or **visual** command at all. IEEE Std. 1003.1-200x extends  
 16799 historical practice, requiring that quoting behavior be made consistent across all *ex* commands,

except for the **map**, **unmap**, **abbreviate**, and **unabbreviate** commands, which historically used `<control>-V` instead of backslashes for quoting. For those four commands, IEEE Std. 1003.1-200x requires conformance to historical practice.

Backslash quoting in *ex* is non-intuitive. Backslash escapes are ignored unless they escape a special character; for example, when performing *file* argument expansion, the string `"\\\"` is equivalent to `'\%'`, not `"\<current path name>"`. This can be confusing for users because backslash is usually one of the characters that causes shell expansion to be performed, and therefore shell quoting rules must be taken into consideration. Generally, quoting characters are only considered if they escape a special character, and a quoting character must be provided for each layer of parsing for which the character is special. As another example, only a single backslash is necessary for the `'\1'` sequence in substitute replacement patterns, because the character `'1'` is not special to any parsing layer above it.

`<control>-V` quoting in *ex* is slightly different from backslash quoting. In the four commands where `<control>-V` quoting applies (**abbreviate**, **unabbreviate**, **map**, and **unmap**), any character may be escaped by a `<control>-V` whether it would have a special meaning or not. IEEE Std. 1003.1-200x requires conformance to historical practice.

Historical implementations of the editor did not require delimiters within character classes to be escaped; for example, the command `:s/[/]//` on the string `"xxx/yyy"` would delete the `'/'` from the string. IEEE Std. 1003.1-200x disallows this historical practice for consistency and because it places a large burden on implementations by requiring that knowledge of regular expressions be built into the editor parser.

Historically, quoting `<newline>` characters in *ex* commands was handled inconsistently. In most cases, the `<newline>` always terminated the command, regardless of any preceding escape character, because backslash characters did not escape `<newline>` characters for most *ex* commands. However, some *ex* commands (for example, **s**, **map**, and **abbreviation**) permitted `<newline>` characters to be escaped (although in the case of **map** and **abbreviation**, `<control>-V` characters escaped them instead of backslashes). This was true in not only the command line, but also **.exrc** and **sourced** files. For example, the command:

```
map = foo<control-V><newline>bar
```

would succeed, although it was sometimes difficult to get the `<control>-V` and the inserted `<newline>` passed to the *ex* parser. For consistency and simplicity of specification, IEEE Std. 1003.1-200x requires that it be possible to escape `<newline>` characters in *ex* commands at all times, using backslashes for most *ex* commands, and using `<control>-V` characters for the **map** and **abbreviation** commands. For example, the command **print<newline>list** is required to be parsed as the single command **print<newline>list**. While this differs from historical practice, IEEE Std. 1003.1-200x developers believed it unlikely that any script or user depended on the historical behavior.

Historically, an error in a command specified using the `-c` option did not cause the rest of the `-c` commands to be discarded. IEEE Std. 1003.1-200x disallows this for consistency with mapped keys, the **@**, **global**, **source**, and **v** commands, the *EXINIT* environment variable, and the **.exrc** files.

## Input Editing in ex

One of the common uses of the historical *ex* editor is over slow network connections. Editors that run in canonical mode can require far less traffic to and from, and far less processing on, the host machine, as well as more easily supporting block-mode terminals. For these reasons, IEEE Std. 1003.1-200x requires that *ex* be implemented using canonical mode input processing, as was done historically.

IEEE Std. 1003.1-200x does not require the historical 4 BSD input editing characters “word erase” or “literal next”. For this reason, it is unspecified how they are handled by *ex*, although they must have the required effect. Implementations that resolve them after the line has been ended using a <newline> or <control>-M character, and implementations that rely on the underlying system terminal support for this processing, are both conforming. Implementations are strongly urged to use the underlying system functionality, if at all possible, for compatibility with other system text input interfaces.

Historically, when the *eof* character was used to decrement the **autoindent** level, the cursor moved to display the new end of the **autoindent** characters, but did not move the cursor to a new line, nor did it erase the <control>-D character from the line. IEEE Std. 1003.1-200x does not specify that the cursor remain on the same line or that the rest of the line is erased; however, implementations are strongly encouraged to provide the best possible user interface; that is, the cursor should remain on the same line, and any <control>-D character on the line should be erased.

IEEE Std. 1003.1-200x does not require the historical 4 BSD input editing character “reprint”, traditionally <control>-R, which redisplayed the current input from the user. For this reason, and because the functionality cannot be implemented after the line has been terminated by the user, IEEE Std. 1003.1-200x makes no requirements about this functionality. Implementations are strongly urged to make this historical functionality available, if possible.

Historically, <control>-Q did not perform a literal next function in *ex*, as it did in *vi*. IEEE Std. 1003.1-200x requires conformance to historical practice to avoid breaking historical *ex* scripts and **.exrc** files.

## eof

Whether the *eof* character immediately modifies the **autoindent** characters in the prompt is left unspecified so that implementations can conform in the presence of systems that do not support this functionality. Implementations are encouraged to modify the line and redisplay it immediately, if possible.

The specification of the handling of the *eof* character differs from historical practice only in that *eof* characters are not discarded if they follow normal characters in the text input. Historically, they were always discarded.

## Command Descriptions in ex

Historically, several commands (for example, **global**, **v**, **visual**, **s**, **write**, **wq**, **yank**, **!**, **<**, **>**, **&**, and **→**) were executable in empty files (that is, the default address(es) were 0), or permitted explicit addresses of 0 (for example, 0 was a valid address, or 0,0 was a valid range). Addresses of 0, or command execution in an empty file, make sense only for commands that add new text to the edit buffer or write commands (because users may wish to write empty files). IEEE Std. 1003.1-200x requires this behavior for such commands and disallows it otherwise, for consistency and simplicity of specification.

A count to an *ex* command has been historically corrected to be no greater than the last line in a file; for example, in a five-line file, the command **1,6print** would fail, but the command **1print300**

16887 would succeed. IEEE Std. 1003.1-200x requires conformance to historical practice.

16888 Historically, the use of flags in *ex* commands could be obscure. General historical practice was as  
 16889 described by IEEE Std. 1003.1-200x, but there were some special cases. For example, the **list**,  
 16890 **number**, and **print** commands ignored trailing address offsets; for example, **3p** +++# would  
 16891 display line 3, and 3 would be the current line after the execution of the command. The **open** and  
 16892 **visual** commands ignored both the trailing offsets and the trailing flags. Also, flags specified to  
 16893 the **open** and **visual** commands interacted badly with the **list** edit option, and setting and then  
 16894 unsetting it during the open/visual session would cause *vi* to stop displaying lines in the  
 16895 specified format. For consistency and simplicity of specification, IEEE Std. 1003.1-200x does not  
 16896 permit any of these exceptions to the general rule.

16897 IEEE Std. 1003.1-200x uses the word *copy* in several places when discussing buffers. This is not  
 16898 intended to imply implementation.

16899 Historically, *ex* users could not specify numeric buffers because of the ambiguity this would  
 16900 cause; for example, in the command **3 delete 2**, it is unclear whether 2 is a buffer name or a  
 16901 *count*. IEEE Std. 1003.1-200x requires conformance to historical practice by default, but does not  
 16902 preclude extensions.

16903 Historically, the contents of the unnamed buffer were frequently discarded after commands that  
 16904 did not explicitly affect it; for example, when using the **edit** command to switch files. For  
 16905 consistency and simplicity of specification, IEEE Std. 1003.1-200x does not permit this behavior.

16906 The *ex* utility did not historically have access to the numeric buffers, and, furthermore, deleting  
 16907 lines in *ex* did not modify their contents. For example, if, after doing a delete in *vi*, the user  
 16908 switched to *ex*, did another delete, and then switched back to *vi*, the contents of the numeric  
 16909 buffers would not have changed. IEEE Std. 1003.1-200x requires conformance to historical  
 16910 practice. Numeric buffers are described in the *ex* utility in order to confine the description of  
 16911 buffers to a single location in IEEE Std. 1003.1-200x.

16912 The metacharacters that trigger shell expansion in *file* arguments match historical practice, as  
 16913 does the method for doing shell expansion. Implementations wishing to provide users with the  
 16914 flexibility to alter the set of metacharacters are encouraged to provide a **shellmeta** string edit  
 16915 option.

16916 Historically, *ex* commands executed from *vi* refreshed the screen when it did not strictly need to  
 16917 do so; for example, **!date > /dev/null** does not require a screen refresh because the output of the  
 16918 UNIX *date* command requires only a single line of the screen. IEEE Std. 1003.1-200x requires that  
 16919 the screen be refreshed if it has been overwritten, but makes no requirements as to how an  
 16920 implementation should make that determination. Implementations may prompt and refresh the  
 16921 screen regardless.

16922 **Abbreviate**

16923 Historical practice was that characters that were entered as part of an abbreviation replacement  
 16924 were subject to **map** expansions, the **showmatch** edit option, further abbreviation expansions,  
 16925 and so on; that is, they were logically pushed onto the terminal input queue, and were not a  
 16926 simple replacement. IEEE Std. 1003.1-200x requires conformance to historical practice.  
 16927 Historical practice was that whenever a non-word character (that had not been escaped by a  
 16928 <control>-V) was entered after a word character, *vi* would check for abbreviations. The check  
 16929 was based on the type of the character entered before the word character of the word/non-word  
 16930 pair that triggered the check. The word character of the word/non-word pair that triggered the  
 16931 check and all characters entered before the trigger pair that were of that type were included in  
 16932 the check, with the exception of <blank> characters, which always delimited the abbreviation.

This means that, for the abbreviation to work, the *lhs* must end with a word character, there can be no transitions from word to non-word characters (or *vice versar*) other than between the last and next-to-last characters in the *lhs*, and there can be no <blank> characters in the *lhs*. In addition, because of the historical quoting rules, it was impossible to enter a literal <control>-V in the *lhs*. IEEE Std. 1003.1-200x requires conformance to historical practice. Historical implementations did not inform users when abbreviations that could never be used were entered; implementations are strongly encouraged to do so.

For example, the following abbreviations will work:

```
:ab (p REPLACE
:ab p REPLACE
:ab ((p REPLACE
```

The following abbreviations will not work:

```
:ab (REPLACE
:ab (pp REPLACE
```

Historical practice is that words on the *vi* colon command line were subject to abbreviation expansion, including the arguments to the **abbrev** (and more interestingly) the **unabbrev** command. Because there are implementations that do not do abbreviation expansion for the first argument to those commands, this is permitted, but not required, by IEEE Std. 1003.1-200x. However, the following sequence:

```
:ab foo bar
:ab foo baz
```

resulted in the addition of an abbreviation of "baz" for the string "bar" in historical *ex/vi*, and the sequence:

```
:ab foo1 bar
:ab foo2 bar
:unabbreviate foo2
```

deleted the abbreviation "foo1", not "foo2". These behaviors are not permitted by IEEE Std. 1003.1-200x because they clearly violate the expectations of the user.

It was historical practice that <control>-V, not backslash, characters be interpreted as escaping subsequent characters in the **abbreviate** command. IEEE Std. 1003.1-200x requires conformance to historical practice; however, it should be noted that an abbreviation containing a <blank> will never work.

## Append

Historically, any text following a vertical-line command separator after an **append**, **change**, or **insert** command became part of the insert text. For example, in the command:

```
:g/pattern/append|stuff1
```

a line containing the text "stuff1" would be appended to each line matching pattern. It was also historically valid to enter:

```
:append|stuff1
stuff2
.
```

and the text on the *ex* command line would be appended along with the text inserted after it. There was an historical bug, however, that the user had to enter two terminating lines (the ' .' lines) to terminate text input mode in this case. IEEE Std. 1003.1-200x requires conformance to



16977 historical practice, but disallows the historical need for multiple terminating lines.

## 16978 **Change**

16979 See the RATIONALE for the **append** command. Historical practice for cursor positioning after  
16980 the change command when no text is input, is as described in IEEE Std. 1003.1-200x. However,  
16981 one System V implementation is known to have been modified such that the cursor is positioned  
16982 on the first address specified, and not on the line before the first address. IEEE Std. 1003.1-200x  
16983 disallows this modification for consistency.

16984 Historically, the **change** command did not support buffer arguments, although some  
16985 implementations allow the specification of an optional buffer. This behavior is neither required  
16986 nor disallowed by IEEE Std. 1003.1-200x.

## 16987 **Change Directory**

16988 A common extension in *ex* implementations is to use the elements of a **cdpath** edit option as  
16989 prefix directories for *path* arguments to **chdir** that are relative path names and that do not have  
16990 ' . ' or " . . " as their first component. Elements in the **cdpath** edit option are colon-separated.  
16991 The initial value of the **cdpath** edit option is the value of the shell *CDPATH* environment  
16992 variable. This feature was not included in IEEE Std. 1003.1-200x because it does not exist in any  
16993 of the implementations considered historical practice.

## 16994 **Copy**

16995 Historical implementations of *ex* permitted copies to lines inside of the specified range; for  
16996 example, **:2,5copy3** was a valid command. IEEE Std. 1003.1-200x requires conformance to  
16997 historical practice.

## 16998 **Delete**

16999 IEEE Std. 1003.1-200x requires support for the historical parsing of a **delete** command followed  
17000 by flags, without any intervening <blank> characters. For example:

17001 **1dp** Deletes the first line and prints the line that was second.

17002 **1delep** As for **1dp**.

17003 **1d** Deletes the first line, saving it in buffer *p*.

17004 **1d p11** (Pee-one-ell.) Deletes the first line, saving it in buffer *p*, and listing the line that was  
17005 second.

## 17006 **Edit**

17007 Historically, any *ex* command could be entered as a *+command* argument to the **edit** command,  
17008 although some (for example, **insert** and **append**) were known to confuse historical  
17009 implementations. For consistency and simplicity of specification, IEEE Std. 1003.1-200x requires  
17010 that any command be supported as an argument to the **edit** command.

17011 Historically, the command argument was executed with the current line set to the last line of the  
17012 file, regardless of whether the **edit** command was executed from visual mode or not.  
17013 IEEE Std. 1003.1-200x requires conformance to historical practice.

17014 Historically, the *+command* specified to the **edit** and **next** commands was delimited by the first  
17015 <blank> character, and there was no way to quote them. For consistency, IEEE Std. 1003.1-200x  
17016 requires that the usual *ex* backslash quoting be provided.

17017 Historically, specifying the *+command* argument to the edit command required a file name to be  
 17018 specified as well; for example, **:edit +100** would always fail. For consistency and simplicity of  
 17019 specification, IEEE Std. 1003.1-200x does not permit this usage to fail for that reason.

17020 Historically, only the cursor position of the last file edited was remembered by the editor.  
 17021 IEEE Std. 1003.1-200x requires that this be supported; however, implementations are permitted  
 17022 to remember and restore the cursor position for any file previously edited.

## 17023 **File**

17024 Historical versions of the *ex* editor **file** command displayed a current line and number of lines in  
 17025 the edit buffer of 0 when the file was empty, while the *vi* **<control>-G** command displayed a  
 17026 current line and number of lines in the edit buffer of 1 in the same situation.  
 17027 IEEE Std. 1003.1-200x does not permit this discrepancy, instead requiring that a message be  
 17028 displayed indicating that the file is empty.

## 17029 **Global**

17030 The two-pass operation of the **global** and **v** commands is not intended to imply implementation,  
 17031 only the required result of the operation.

17032 The current line and column are set as specified for the individual *ex* commands. This  
 17033 requirement is cumulative; that is, the current line and column must track across all the  
 17034 commands executed by the **global** or **v** commands.

## 17035 **Insert**

17036 See the RATIONALE for the **append** command.

17037 Historically, **insert** could not be used with an address of zero; that is, not when the edit buffer  
 17038 was empty. IEEE Std. 1003.1-200x requires that this command behave consistently with the  
 17039 **append** command.

## 17040 **Join**

17041 The action of the **join** command in relation to the special characters is only defined for the  
 17042 POSIX locale because the correct amount of white space after a period varies; in Japanese none is  
 17043 required, in French only a single space, and so on.

## 17044 **List**

17045 The historical output of the **list** command was potentially ambiguous. The standard developers  
 17046 believed correcting this to be more important than adhering to historical practice, and  
 17047 IEEE Std. 1003.1-200x requires unambiguous output.

## 17048 **Map**

17049 Historically, command mode maps only applied to command names; for example, if the  
 17050 character 'x' was mapped to 'y', the command **fx** searched for the 'x' character, not the 'y'  
 17051 character. IEEE Std. 1003.1-200x requires this behavior. Historically, entering **<control>-V** as the  
 17052 first character of a *vi* command was an error. Several implementations have extended the  
 17053 semantics of *vi* such that **<control>-V** means that the subsequent command character is not  
 17054 mapped. This is permitted, but not required, by IEEE Std. 1003.1-200x. Regardless, using  
 17055 **<control>-V** to escape the second or later character in a sequence of characters that might match  
 17056 a **map** command, or any character in text input mode, is historical practice, and stops the entered  
 17057 keys from matching a map. IEEE Std. 1003.1-200x requires conformance to historical practice.

17058 Historical implementations permitted digits to be used as a **map** command *lhs*, but then ignored  
17059 the map. IEEE Std. 1003.1-200x requires that the mapped digits not be ignored.

17060 The historical implementation of the **map** command did not permit **map** commands that were  
17061 more than a single character in length if the first character was printable. This behavior is  
17062 permitted, but not required, by IEEE Std. 1003.1-200x.

17063 Specifications of “function keys” in the **map** command were omitted because the historical  
17064 specification of such was too simple to be generally useful in a portable manner. Historical  
17065 practice is that a ‘#’ followed by a number mapped to that number function key—for example,  
17066 “#3”—was function key 3 for the current terminal, as well as being accessible using the keys  
17067 ‘#’ and ‘3’. Implementations have extended this semantic to permit users to specify things  
17068 like “#up” and “#page\_forward” as well. These extensions are permitted, but not required,  
17069 by IEEE Std. 1003.1-200x.

17070 Historically, mapped characters were remapped unless the **remap** edit option was not set, or the  
17071 prefix of the mapped characters matched the mapping characters; for example, in the **map**:

17072 :map ab abcd

17073 the characters “ab” were used as is and were not remapped, but the characters “cd” were  
17074 mapped if appropriate. This can cause infinite loops in the *vi* mapping mechanisms.  
17075 IEEE Std. 1003.1-200x requires conformance to historical practice, and that such loops be  
17076 interruptible.

17077 Text input maps had the same problems with expanding the *lhs* for the **ex map!** and **unmap!**  
17078 command as did the **ex abbreviate** and **unabbreviate** commands. See the RATIONALE for the **ex**  
17079 **abbreviate** command. IEEE Std. 1003.1-200x requires similar modification of some historical  
17080 practice for the **map** and **unmap** commands, as described for the **abbreviate** and **unabbreviate**  
17081 commands.

17082 Historically, **maps** that were subsets of other **maps** behaved differently depending on the order  
17083 in which they were defined. For example:

17084 :map! ab short  
17085 :map! abc long

17086 would always translate the characters “ab” to “short”, regardless of how fast the characters  
17087 “abc” were entered. If the entry order was reversed:

17088 :map! abc long  
17089 :map! ab short

17090 the characters “ab” would cause the editor to pause, waiting for the completing ‘c’ character,  
17091 and the characters might never be mapped to “short”. For consistency and simplicity of  
17092 specification, IEEE Std. 1003.1-200x requires that the shortest match be used at all times.

17093 The length of time the editor spends waiting for the characters to complete the *lhs* is unspecified  
17094 because the timing capabilities of systems are often inexact and variable, and it may depend on  
17095 other factors such as the speed of the connection. The time should be long enough for the user to  
17096 be able to complete the sequence, but not long enough for the user to have to wait. Some  
17097 implementations of *vi* have added a **keytime** option, which permits users to set the number of  
17098 0,1 seconds the editor waits for the completing characters. Because mapped terminal function  
17099 and cursor keys tend to start with an <ESC> character, and <ESC> is the key ending *vi* text input  
17100 mode, **maps** starting with <ESC> characters are generally exempted from this timeout period,  
17101 or, at least timed out differently.

17102 **Mark**

17103 Historically, users were able to set the “previous context” marks explicitly. In addition, the *ex*  
 17104 commands ” and ” and the *vi* commands ”, “, “, and ” all referred to the same mark. In addition,  
 17105 the previous context marks were not set if the command, with which the address setting the  
 17106 mark was associated, failed. IEEE Std. 1003.1-200x requires conformance to historical practice.  
 17107 Historically, if marked lines were deleted, the mark was also deleted, but would reappear if the  
 17108 change was undone. IEEE Std. 1003.1-200x requires conformance to historical practice.

17109 The description of the special events that set the ‘ and ’ marks matches historical practice. For  
 17110 example, historically the command */a/,b/* did not set the ‘ and ’ marks, but the command  
 17111 */a/,b/delete* did.

17112 **Next**

17113 Historically, any *ex* command could be entered as a *+command* argument to the **next** command,  
 17114 although some (for example, **insert** and **append**) were known to confuse historical  
 17115 implementations. IEEE Std. 1003.1-200x requires that any command be permitted and that it  
 17116 behave as specified. The **next** command can accept more than one file, so usage such as:

17117 `next `ls [abc] ``

17118 is valid; it need not be valid for the **edit** or **read** commands, for example, because they expect  
 17119 only one file name.

17120 Historically, the **next** command behaved differently from the **:rewind** command in that it  
 17121 ignored the force flag if the **autowrite** flag was set. For consistency, IEEE Std. 1003.1-200x does  
 17122 not permit this behavior.

17123 Historically, the **next** command positioned the cursor as if the file had never been edited before,  
 17124 regardless. IEEE Std. 1003.1-200x does not permit this behavior, for consistency with the **edit**  
 17125 command.

17126 Implementations wanting to provide a counterpart to the **next** command that edited the  
 17127 previous file have used the command **prev[ious]**, which takes no *file* argument.  
 17128 IEEE Std. 1003.1-200x does not require this command.

17129 **Open**

17130 Historically, the **open** command would fail if the **open** edit option was not set.  
 17131 IEEE Std. 1003.1-200x does not mention the **open** edit option and does not require this behavior.  
 17132 Some historical implementations do not permit entering open mode from open or visual mode,  
 17133 only from *ex* mode. For consistency, IEEE Std. 1003.1-200x does not permit this behavior.

17134 Historically, entering open mode from the command line (that is, *vi +open*) resulted in  
 17135 anomalous behaviors; for example, the *ex* file and *set* commands, and the *vi* command  
 17136 <control>-G did not work. For consistency, IEEE Std. 1003.1-200x does not permit this behavior.

17137 Historically, the **open** command only permitted ‘ / ’ characters to be used as the search pattern  
 17138 delimiter. For consistency, IEEE Std. 1003.1-200x requires that the search delimiters used by the  
 17139 **s**, **global**, and **v** commands be accepted as well.

17140 **Preserve**

17141 The **preserve** command does not historically cause the file to be considered unmodified for the  
 17142 purposes of future commands that may exit the editor. IEEE Std. 1003.1-200x requires  
 17143 conformance to historical practice.

17144 Historical documentation stated that mail was not sent to the user when preserve was executed;  
 17145 however, historical implementations did send mail in this case. IEEE Std. 1003.1-200x requires  
 17146 conformance to the historical implementations.

17147 **Print**

17148 The writing of NUL by the **print** command is not specified as a special case because the standard  
 17149 developers did not want to require **ex** to support NUL characters. Historically, characters were  
 17150 displayed using the ARPA standard mappings, which are as follows:

- 17151 1. Printable characters are left alone.
- 17152 2. Control characters less than \177 are represented as '^' followed by the character offset  
 17153 from the '@' character in the ASCII map; for example, \007 is represented as '^G'.
- 17154 3. \177 is represented as '^' followed by '?'.

17155 The display of characters having their eighth bit set was less standard. Existing implementations  
 17156 use hex (0x00), octal (\000), and a meta-bit display. (The latter displayed bytes that had their  
 17157 eighth bit set as the two characters "M-" followed by the seven-bit display as described above.)  
 17158 The latter probably has the best claim to historical practice because it was used for the -v option  
 17159 of 4 BSD and 4 BSD-derived versions of the **cat** utility since 1980.

17160 No specific display format is required by IEEE Std. 1003.1-200x.

17161 Explicit dependence on the ASCII character set has been avoided where possible, hence the use  
 17162 of the phrase an "implementation-dependent multi-character sequence" for the display of non-  
 17163 printable characters in preference to the historical usage of, for instance, "^I" for the <tab>  
 17164 character. Implementations are encouraged to conform to historical practice in the absence of  
 17165 any strong reason to diverge.

17166 Historically, all **ex** commands beginning with the letter 'p' could be entered using capitalized  
 17167 versions of the commands; for example, **P[rint]**, **Pre[serve]**, and **Pu[t]** were all valid command  
 17168 names. IEEE Std. 1003.1-200x permits, but does not require, this historical practice because  
 17169 capital forms of the commands are used by some implementations for other purposes.

17170 **Put**

17171 Historically, an **ex put** command, executed from open or visual mode, was the same as the open  
 17172 or visual mode **P** command, if the buffer was named and was cut in character mode, and the  
 17173 same as the **p** command if the buffer was named and cut in line mode. If the unnamed buffer  
 17174 was the source of the text, the entire line from which the text was taken was usually **put**, and the  
 17175 buffer was handled as if in line mode, but it was possible to get extremely anomalous behavior.  
 17176 In addition, using the **Q** command to switch into **ex** mode, and then doing a **put** often resulted in  
 17177 errors as well, such as appending text that was unrelated to the (supposed) contents of the  
 17178 buffer. For consistency and simplicity of specification, IEEE Std. 1003.1-200x does not permit  
 17179 these behaviors. All **ex put** commands are required to operate in line mode, and the contents of  
 17180 the buffers are not altered by changing the mode of the editor.

17181 **Read**

17182 Historically, an **ex read** command executed from open or visual mode, executed in an empty file,  
 17183 left an empty line as the first line of the file. For consistency and simplicity of specification,  
 17184 IEEE Std. 1003.1-200x does not permit this behavior. Historically, a **read** in open or visual mode  
 17185 from a program left the cursor at the last line read in, not the first. For consistency,  
 17186 IEEE Std. 1003.1-200x does not permit this behavior.

17187 Historical implementations of **ex** were unable to undo **read** commands that read from the output  
 17188 of a program. For consistency, IEEE Std. 1003.1-200x does not permit this behavior.

17189 Historically, the **ex** and **vi** message after a successful **read** or **write** command specified  
 17190 “characters”, not “bytes”. IEEE Std. 1003.1-200x requires that the number of bytes be displayed,  
 17191 not the number of characters, because it may be difficult in multi-byte implementations to  
 17192 determine the number of characters read. Implementations are encouraged to clarify the  
 17193 message displayed to the user.

17194 Historically, reads were not permitted on files other than type regular, except that FIFO files  
 17195 could be read (probably only because they did not exist when **ex** and **vi** were originally written).  
 17196 Because the historical **ex** evaluated **read!** and **read !** equivalently, there can be no optional way  
 17197 to force the read. IEEE Std. 1003.1-200x permits, but does not require, this behavior.

17198 **Recover**

17199 Some historical implementations of the editor permitted users to recover the edit buffer contents  
 17200 from a previous edit session, and then exit without saving those contents (or explicitly  
 17201 discarding them). The intent of IEEE Std. 1003.1-200x in requiring that the edit buffer be treated  
 17202 as already modified is to prevent this user error.

17203 **Rewind**

17204 Historical implementations supported the **rewind** command when the user was editing the first  
 17205 file in the list; that is, the file that the **rewind** command would edit. IEEE Std. 1003.1-200x  
 17206 requires conformance to historical practice.

17207 **Substitute**

17208 Historically, **ex** accepted an **r** option to the **s** command. The effect of the **r** option was to use the  
 17209 last regular expression used in any command as the pattern, the same as the **~** command. The **r**  
 17210 option is not required by IEEE Std. 1003.1-200x. Historically, the **c** and **g** options were toggled;  
 17211 for example, the command **:s/abc/def/** was the same as **s/abc/def/ccccgggg**. For simplicity of  
 17212 specification, IEEE Std. 1003.1-200x does not permit this behavior.

17213 The tilde command is often used to replace the last search RE. For example, in the sequence:

```
17214 s/red/blue/
17215 /green
17216 ~
```

17217 the **~** command is equivalent to:

```
17218 s/green/blue/
```

17219 Historically, **ex** accepted all of the following forms:

```
17220 s/abc/def/
17221 s/abc/def
17222 s/abc/
17223 s/abc
```

|       |                                                                                                                          |
|-------|--------------------------------------------------------------------------------------------------------------------------|
| 17224 | IEEE Std. 1003.1-200x requires conformance to this historical practice.                                                  |
| 17225 | The <b>s</b> command presumes that the '^' character only occupies a single column in the display.                       |
| 17226 | Much of the <i>ex</i> and <i>vi</i> specification presumes that the <space> character only occupies a single             |
| 17227 | column in the display. There are no known character sets for which this is not true.                                     |
| 17228 | Historically, the final column position for the substitute commands was based on previous                                |
| 17229 | column movements; a search for a pattern followed by a substitution would leave the column                               |
| 17230 | position unchanged, while a 0 command followed by a substitution would change the column                                 |
| 17231 | position to the first non-<blank>. For consistency and simplicity of specification,                                      |
| 17232 | IEEE Std. 1003.1-200x requires that the final column position always be set to the first non-                            |
| 17233 | <blank>.                                                                                                                 |
| 17234 | <b>Set</b>                                                                                                               |
| 17235 | Historical implementations redisplayed all of the options for each occurrence of the <b>all</b> keyword.                 |
| 17236 | IEEE Std. 1003.1-200x permits, but does not require, this behavior.                                                      |
| 17237 | <b>Tag</b>                                                                                                               |
| 17238 | No requirement is made as to where <i>ex</i> and <i>vi</i> shall look for the file referenced by the tag entry.          |
| 17239 | Historical practice has been to look for the path found in the tags file, based on the current                           |
| 17240 | directory. A useful extension found in some implementations is to look based on the directory                            |
| 17241 | containing the tags file that held the entry, as well. No requirement is made as to which                                |
| 17242 | reference for the tag in the tags file is used. This is deliberate, in order to permit extensions such                   |
| 17243 | as multiple entries in a tags file for a tag.                                                                            |
| 17244 | Because users often specify many different tags files, some of which need not be relevant or exist                       |
| 17245 | at any particular time, IEEE Std. 1003.1-200x requires that error messages about problem tags                            |
| 17246 | files be displayed only if the requested tag is not found, and then, only once for each time that                        |
| 17247 | the <b>tag</b> edit option is changed.                                                                                   |
| 17248 | The requirement that the current edit buffer be unmodified is only necessary if the file indicated                       |
| 17249 | by the tag entry is not the same as the current file (as defined by the current path name).                              |
| 17250 | Historically, the file would be reloaded if the file name had changed, as well as if the file name                       |
| 17251 | was different from the current path name. For consistency and simplicity of specification,                               |
| 17252 | IEEE Std. 1003.1-200x does not permit this behavior, requiring that the name be the only factor in                       |
| 17253 | the decision.                                                                                                            |
| 17254 | Historically, <i>vi</i> only searched for tags in the current file from the current cursor to the end of the             |
| 17255 | file, and therefore, if the <b>wraps</b> option was not set, tags occurring before the current cursor                    |
| 17256 | were not found. IEEE Std. 1003.1-200x considers this a bug, and implementations are required to                          |
| 17257 | search for the first occurrence in the file, regardless.                                                                 |
| 17258 | <b>Undo</b>                                                                                                              |
| 17259 | The <b>undo</b> description deliberately uses the word “modified”. The <b>undo</b> command is not                        |
| 17260 | intended to undo commands that replace the contents of the edit buffer, such as <b>edit</b> , <b>next</b> , <b>tag</b> , |
| 17261 | or <b>recover</b> .                                                                                                      |
| 17262 | Cursor positioning after the <b>undo</b> command was inconsistent in the historical <i>vi</i> , sometimes                |
| 17263 | attempting to restore the original cursor position ( <b>global</b> , <b>undo</b> , and <b>v</b> commands), and           |
| 17264 | sometimes, in the presence of maps, placing the cursor on the last line added or changed instead                         |
| 17265 | of the first. IEEE Std. 1003.1-200x requires a simplified behavior for consistency and simplicity of                     |
| 17266 | specification.                                                                                                           |

17267 **Version**

17268 The **version** command cannot be exactly specified since there is no widely-accepted definition of  
 17269 what the version information should contain. Implementations are encouraged to do something  
 17270 reasonably intelligent.

17271 **Write**

17272 Historically, the *ex* and *vi* message after a successful **read** or **write** command specified  
 17273 “characters”, not “bytes”. IEEE Std. 1003.1-200x requires that the number of bytes be displayed,  
 17274 not the number of characters because it may be difficult in multi-byte implementations to  
 17275 determine the number of characters written. Implementations are encouraged to clarify the  
 17276 message displayed to the user.

17277 Implementation-dependent tests are permitted so that implementations can make additional  
 17278 checks; for example, for locks or file modification times.

17279 Historically, attempting to append to a nonexistent file caused an error. It has been left  
 17280 unspecified in IEEE Std. 1003.1-200x to permit implementations to let the **write** succeed, so that  
 17281 the append semantics are similar to those of the historical *csk*.

17282 Historical *vi* permitted empty edit buffers to be written. However, since the way *vi* got around  
 17283 dealing with “empty” files was to always have a line in the edit buffer, no matter what, it wrote  
 17284 them as files of a single, empty line. IEEE Std. 1003.1-200x does not permit this behavior.

17285 Historically, *ex* restored standard output and standard error to their values as of when *ex* was  
 17286 invoked, before writes to programs were performed. This could disturb the terminal  
 17287 configuration as well as be a security issue for some terminals. IEEE Std. 1003.1-200x does not  
 17288 permit this, requiring that the program output be captured and displayed as if by the *ex* **print**  
 17289 command.

17290 **Adjust Window**

17291 Historically, the line count was set to the value of the **scroll** option if the type character was  
 17292 end-of-file. This feature was broken on most historical implementations long ago, however, and  
 17293 is not documented anywhere. For this reason, IEEE Std. 1003.1-200x is resolutely silent.

17294 Historically, the **z** command was <blank> character-sensitive and **z +** and **z -** did different  
 17295 things than **z+** and **z-** because the type could not be distinguished from a flag. (The commands  
 17296 **z .** and **z =** were historically invalid.) IEEE Std. 1003.1-200x requires conformance to this  
 17297 historical practice.

17298 Historically, the **z** command was further <blank> character-sensitive in that the *count* could not  
 17299 be <blank> character-delimited; for example, the commands **z= 5** and **z- 5** were also invalid.  
 17300 Because the *count* is not ambiguous with respect to either the type character or the flags, this is  
 17301 not permitted by IEEE Std. 1003.1-200x.

17302 **Escape**

17303 Historically, *ex* filter commands only read the standard output of the commands, letting  
 17304 standard error appear on the terminal as usual. The *vi* utility, however, read both standard  
 17305 output and standard error. IEEE Std. 1003.1-200x requires the latter behavior for both *ex* and *vi*,  
 17306 for consistency.



### 17307 **Shift Left and Shift Right**

17308 Historically, it was possible to add shift characters to increase the effect of the command; for  
 17309 example, <<< outdented (or >>> indented) the lines 3 levels of indentation instead of the default  
 17310 1. IEEE Std. 1003.1-200x requires conformance to historical practice.

### 17311 **<control>-D**

17312 Historically, the <control>-D command erased the prompt, providing the user with an unbroken  
 17313 presentation of lines from the edit buffer. This is not required by IEEE Std. 1003.1-200x;  
 17314 implementations are encouraged to provide it if possible. Historically, the <control>-D  
 17315 command took, and then ignored, a *count*. IEEE Std. 1003.1-200x does not permit this behavior.

### 17316 **Write Line Number**

17317 Historically, the *ex* = command, when executed in *ex* mode in an empty edit buffer, reported 0,  
 17318 and from open or visual mode, reported 1. For consistency and simplicity of specification,  
 17319 IEEE Std. 1003.1-200x does not permit this behavior.

### 17320 **Execute**

17321 Historically, *ex* did not correctly handle the inclusion of text input commands (that is, **append**,  
 17322 **insert**, and **change**) in executed buffers. IEEE Std. 1003.1-200x does not permit this exclusion for  
 17323 consistency.

17324 Historically, the logical contents of the buffer being executed did not change if the buffer itself  
 17325 were modified by the commands being executed; that is, buffer execution did not support self-  
 17326 modifying code. IEEE Std. 1003.1-200x requires conformance to historical practice.

17327 Historically, the @ command took a range of lines, and the @ buffer was executed once per line,  
 17328 with the current line ( ' . ' ) set to each specified line. IEEE Std. 1003.1-200x requires conformance  
 17329 to historical practice.

17330 Some historical implementations did not notice if errors occurred during buffer execution. This,  
 17331 coupled with the ability to specify a range of lines for the *ex* @ command, makes it trivial to  
 17332 cause them to drop core. IEEE Std. 1003.1-200x requires that implementations stop buffer  
 17333 execution if any error occurs, if the specified line doesn't exist, or if the contents of the edit buffer  
 17334 itself are replaced (for example, the buffer executes the *ex* :**edit** command).

### 17335 **Regular Expressions in ex**

17336 Historical practice is that the characters in the replacement part of the last *s* command—that is,  
 17337 those matched by entering a '~' in the regular expression—were not further expanded by the  
 17338 regular expression engine. So, if the characters contained the string "a . , " they would match  
 17339 'a' followed by " . , " and not 'a' followed by any character. IEEE Std. 1003.1-200x requires  
 17340 conformance to historical practice.

### 17341 **Edit Options in ex**

17342 The following paragraphs describe the historical behavior of some edit options that were not, for  
 17343 whatever reason, included in IEEE Std. 1003.1-200x. Implementations are strongly encouraged  
 17344 to only use these names if the functionality described here is fully supported.

17345 **extended** The **extended** edit option has been used in some implementations of *vi* to provide  
 17346 extended regular expressions instead of basic regular expressions. This option was  
 17347 omitted from IEEE Std. 1003.1-200x because it is not widespread historical practice.

|       |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------|------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 17348 | <b>flash</b>     | The <b>flash</b> edit option historically caused the screen to flash instead of beeping on error. This option was omitted from IEEE Std. 1003.1-200x because it is not found in some historical implementations.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 17349 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17350 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17351 | <b>hardtabs</b>  | The <b>hardtabs</b> edit option historically defined the number of columns between hardware tab settings. This option was omitted from IEEE Std. 1003.1-200x because it was believed to no longer be generally useful.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 17352 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17353 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17354 | <b>modeline</b>  | The <b>modeline</b> (sometimes named <b>modelines</b> ) edit option historically caused <i>ex</i> or <i>vi</i> to read the five first and last lines of the file for editor commands. This option is a security problem, and vendors are strongly encouraged to delete it from historical implementations.                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 17355 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17356 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17357 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17358 | <b>open</b>      | The <b>open</b> edit option historically disallowed the <i>ex</i> <b>open</b> and <b>visual</b> commands. This edit option was omitted because these commands are required by IEEE Std. 1003.1-200x.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 17359 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17360 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17361 | <b>optimize</b>  | The <b>optimize</b> edit option historically expedited text throughput by setting the terminal to not do automatic carriage returns when printing more than one logical line of output. This option was omitted from IEEE Std. 1003.1-200x because it was intended for terminals without addressable cursors, which are rarely, if ever, still used.                                                                                                                                                                                                                                                                                                                                                                                  |
| 17362 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17363 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17364 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17365 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17366 | <b>ruler</b>     | The <b>ruler</b> edit option has been used in some implementations of <i>vi</i> to present a current row/column ruler for the user. This option was omitted from IEEE Std. 1003.1-200x because it is not widespread historical practice.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 17367 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17368 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17369 | <b>sourceany</b> | The <b>sourceany</b> edit option historically caused <i>ex</i> or <i>vi</i> to source start-up files that were owned by users other than the user running the editor. This option is a security problem, and vendors are strongly encouraged to remove it from their implementations.                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 17370 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17371 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17372 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17373 | <b>timeout</b>   | The <b>timeout</b> edit option historically enabled the (now standard) feature of only waiting for a short period before returning keys that could be part of a macro. This feature was omitted from IEEE Std. 1003.1-200x because its behavior is now standard, it is not widely useful, and it was rarely documented.                                                                                                                                                                                                                                                                                                                                                                                                               |
| 17374 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17375 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17376 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17377 | <b>verbose</b>   | The <b>verbose</b> edit option has been used in some implementations of <i>vi</i> to cause <i>vi</i> to output error messages for common errors; for example, attempting to move the cursor past the beginning or end of the line instead of only alerting the screen. (The historical <i>vi</i> only alerted the terminal and presented no message for such errors. The historical editor option <b>terse</b> did not select when to present error messages, it only made existing error messages more or less verbose.) This option was omitted from IEEE Std. 1003.1-200x because it is not widespread historical practice; however, implementors are encouraged to use it if they wish to provide error messages for naive users. |
| 17378 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17379 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17380 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17381 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17382 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17383 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17384 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17385 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17386 | <b>wraplen</b>   | The <b>wraplen</b> edit option has been used in some implementations of <i>vi</i> to specify an automatic margin measured from the left margin instead of from the right margin. This is useful when multiple screen sizes are being used to edit a single file. This option was omitted from IEEE Std. 1003.1-200x because it is not widespread historical practice; however, implementors are encouraged to use it if they add this functionality.                                                                                                                                                                                                                                                                                  |
| 17387 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17388 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17389 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17390 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 17391 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

17392      **autoindent, ai**

17393      Historically, the command **0a** did not do any autoindentation, regardless of the current  
 17394      indentation of line 1. IEEE Std. 1003.1-200x requires that any indentation present in line 1 be  
 17395      used.

17396      **autoprint, ap**

17397      Historically, the **autoprint** edit option was not completely consistent or based solely on  
 17398      modifications to the edit buffer. Exceptions were the **read** command (when reading from a file,  
 17399      but not from a filter), the **append**, **change**, **insert**, **global**, and **v** commands, all of which were not  
 17400      affected by **autoprint**, and the **tag** command, which was affected by **autoprint**.  
 17401      IEEE Std. 1003.1-200x requires conformance to historical practice.

17402      Historically, the **autoprint** option only applied to the last of multiple commands entered using  
 17403      vertical-bar delimiters; for example, **delete** <newline> was affected by **autoprint**, but  
 17404      **delete|version** <newline> was not. IEEE Std. 1003.1-200x requires conformance to historical  
 17405      practice.

17406      **autowrite, aw**

17407      Appending the '!' character to the **ex next** command to avoid performing an automatic write  
 17408      was not supported in historical implementations. IEEE Std. 1003.1-200x requires that the  
 17409      behavior match the other **ex** commands for consistency.

17410      **ignorecase, ic**

17411      Historical implementations of case-insensitive matching (the **ignorecase** edit option) lead to  
 17412      counterintuitive situations when uppercase characters were used in range expressions.  
 17413      Historically, the process was as follows:

- 17414      1. Take a line of text from the edit buffer.
- 17415      2. Convert uppercase to lowercase in text line.
- 17416      3. Convert uppercase to lowercase in regular expressions, except in character class  
 17417      specifications.
- 17418      4. Match regular expressions against text.

17419      This would mean that, with **ignorecase** in effect, the text:

17420      The cat sat on the mat

17421      would be matched by

17422      /^the/

17423      but not by:

17424      /^[A-Z]he/

17425      For consistency with other commands implementing regular expressions, IEEE Std. 1003.1-200x  
 17426      does not permit this behavior.

17427 **paragraphs, para**

17428 Earlier versions of IEEE Std. 1003.1-200x made the default **paragraphs** and **sections** edit options  
 17429 implementation-dependent, arguing they were historically oriented to the UNIX system *troff* text  
 17430 formatter, and a “portable user” could use the { }, [[, ]], (, and ) commands in open or visual  
 17431 mode and have the cursor stop in unexpected places. IEEE Std. 1003.1-200x specifies their values  
 17432 in the POSIX locale because the unusual grouping (they only work when grouped into two  
 17433 characters at a time) means that they cannot be used for general purpose movement, regardless.

17434 **readonly**

17435 Implementations are encouraged to provide the best possible information to the user as to the  
 17436 read-only status of the file, with the exception that they should not consider the current special  
 17437 privileges of the process. This provides users a safety net because they must force the overwrite  
 17438 of read-only files, even when running with additional privileges.

17439 The **readonly** edit option specification largely conforms to historical practice. The only  
 17440 difference is that historical implementations did not notice that the user had set the **readonly**  
 17441 edit option in cases where the file was already marked read-only for some reason, and would  
 17442 therefore reinitialize the **readonly** edit option the next time the contents of the edit buffer were  
 17443 replaced. This behavior is disallowed by IEEE Std. 1003.1-200x.

17444 **report**

17445 The requirement that lines copied to a buffer interact differently than deleted lines is historical  
 17446 practice. For example, if the **report** edit option is set to 3, deleting 3 lines will cause a report to be  
 17447 written, but 4 lines must be copied before a report is written.

17448 The requirement that the **ex global**, **v**, **open**, **undo**, and **visual** commands present reports based  
 17449 on the total number of lines added or deleted during the command execution, and that  
 17450 commands executed by the **global** and **v** commands not present reports, is historical practice.  
 17451 IEEE Std. 1003.1-200x extends historical practice by requiring that buffer execution be treated  
 17452 similarly. The reasons for this are two-fold. Historically, only the report by the last command  
 17453 executed from the buffer would be seen by the user, as each new report would overwrite the  
 17454 last. In addition, the standard developers believed that buffer execution had more in common  
 17455 with **global** and **v** commands than it did with other **ex** commands, and should behave similarly,  
 17456 for consistency and simplicity of specification.

17457 **showmatch, sm**

17458 The length of time the cursor spends on the matching character is unspecified because the  
 17459 timing capabilities of systems are often inexact and variable. The time should be long enough for  
 17460 the user to notice, but not long enough for the user to become annoyed. Some implementations  
 17461 of *vi* have added a **matchtime** option that permits users to set the number of 0,1 second intervals  
 17462 the cursor pauses on the matching character.

17463 **showmode**

17464 The **showmode** option has been used in some historical implementations of *ex* and *vi* to display  
 17465 the current editing mode when in open or visual mode. The editing modes have generally  
 17466 included “command” and “input”, and sometimes other modes such as “replace” and  
 17467 “change”. The string was usually displayed on the bottom line of the screen at the far right-hand  
 17468 corner. In addition, a preceding ‘ ’ character often denoted if the contents of the edit buffer had  
 17469 been modified. The latter display has sometimes been part of the **showmode** option, and  
 17470 sometimes based on another option. This option was not available in the 4 BSD historical  
 17471 implementation of *vi*, but was viewed as generally useful, particularly to novice users, and is

- 17472 required by IEEE Std. 1003.1-200x.
- 17473 The **smd** shorthand for the **showmode** option was not present in all historical implementations  
17474 of the editor. IEEE Std. 1003.1-200x requires it, for consistency.
- 17475 Not all historical implementations of the editor displayed a mode string for command mode,  
17476 differentiating command mode from text input mode by the absence of a mode string.  
17477 IEEE Std. 1003.1-200x permits this behavior for consistency with historical practice, but  
17478 implementations are encouraged to provide a display string for both modes.
- 17479 **slowopen**
- 17480 Historically the **slowopen** option was automatically set if the terminal baud rate was less than  
17481 1 200 baud, or if the baud rate was 1 200 baud and the **redraw** option was not set. The **slowopen**  
17482 option had two effects. First, when inserting characters in the middle of a line, characters after  
17483 the cursor would not be pushed ahead, but would appear to be overwritten. Second, when  
17484 creating a new line of text, lines after the current line would not be scrolled down, but would  
17485 appear to be overwritten. In both cases, ending text input mode would cause the screen to be  
17486 refreshed to match the actual contents of the edit buffer. Finally, terminals that were sufficiently  
17487 intelligent caused the editor to ignore the **slowopen** option. IEEE Std. 1003.1-200x permits most  
17488 historical behavior, extending historical practice to require **slowopen** behaviors if the edit option  
17489 is set by the user.
- 17490 **tags**
- 17491 The default path for tags files is left unspecified as implementations may have their own **tags**  
17492 implementations that do not correspond to the historical ones. The default **tags** option value  
17493 should probably at least include the file **./tags**.
- 17494 **term**
- 17495 Historical implementations of **ex** and **vi** ignored changes to the **term** edit option after the initial  
17496 terminal information was loaded. This is permitted by IEEE Std. 1003.1-200x; however,  
17497 implementations are encouraged to permit the user to modify their terminal type at any time.
- 17498 **terse**
- 17499 Historically, the **terse** edit option optionally provided a shorter, less descriptive error message,  
17500 for some error messages. This is permitted, but not required, by IEEE Std. 1003.1-200x.  
17501 Historically, most common visual mode errors (for example, trying to move the cursor past the  
17502 end of a line) did not result in an error message, but simply alerted the terminal.  
17503 Implementations wishing to provide messages for novice users are urged to do so based on the  
17504 **edit** option **verbose**, and not **terse**.
- 17505 **window**
- 17506 In historical implementations, the default for the **window** edit option was based on the baud  
17507 rate as follows:
- 17508 1. If the baud rate was less than 1 200, the **edit** option **w300** set the window value; for  
17509 example, the line:  
17510 `set w300=12`  
17511 would set the window option to 12 if the baud rate was less than 1 200.
  - 17512 2. If the baud rate was equal to 1 200, the **edit** option **w1200** set the window value.

17513 3. If the baud rate was greater than 1 200, the **edit** option **w9600** set the window value.

17514 The **w300**, **w1200**, and **w9600** options do not appear in IEEE Std. 1003.1-200x because of their  
17515 dependence on specific baud rates.

17516 In historical implementations, the size of the window displayed by various commands was  
17517 related to, but not necessarily the same as, the **window** edit option. For example, the size of the  
17518 window was set by the **ex** command **visual 10**, but it did not change the value of the **window**  
17519 edit option. However, changing the value of the **window** edit option did change the number of  
17520 lines that were displayed when the screen was repainted. IEEE Std. 1003.1-200x does not permit  
17521 this behavior in the interests of consistency and simplicity of specification, and requires that all  
17522 commands that change the number of lines that are displayed do it by setting the value of the  
17523 **window** edit option.

17524 **wrapmargin, wm**

17525 Historically, the **wrapmargin** option did not affect maps inserting characters that also had  
17526 associated *counts*; for example **:map K 5aABC DEF**. Unfortunately, there are widely used  
17527 maps that depend on this behavior. For consistency and simplicity of specification,  
17528 IEEE Std. 1003.1-200x does not permit this behavior.

17529 Historically, **wrapmargin** was calculated using the column display width of all characters on the  
17530 screen. For example, an implementation using "**^I**" to represent <tab> characters when the **list**  
17531 edit option was set, where '**^**' and '**I**' each took up a single column on the screen, would  
17532 calculate the **wrapmargin** based on a value of 2 for each <tab> character. The **number** edit  
17533 option similarly changed the effective length of the line as well. IEEE Std. 1003.1-200x requires  
17534 conformance to historical practice.

17535 **FUTURE DIRECTIONS**

17536 None.

17537 **SEE ALSO**

17538 *ed, sed, stty, vi*, the System Interfaces volume of IEEE Std. 1003.1-200x, *access()*

17539 **CHANGE HISTORY**

17540 First released in Issue 2.

17541 **Issue 4**

17542 Aligned with the ISO/IEC 9945-2: 1993 standard.

17543 **Issue 5**

17544 The FUTURE DIRECTIONS section is added.

17545 **Issue 6**

17546 This utility is now marked as part of the User Portability Utilities option.

17547 The obsolescent SYNOPSIS is removed, removing the *+command* and *-* options.

17548 The following new requirements on POSIX implementations derive from alignment with the  
17549 Single UNIX Specification:

- 17550 • The **-l** option is added.
- 17551 • In the *map* command description, the sequence *#digit* is added.
- 17552 • The **directory**, **edcompatible**, **redraw**, **slowopen**, and **lisp** edit options are added.

17553 The **ex** utility is extensively changed for alignment with the IEEE P1003.2b draft standard. This  
17554 includes changes as a result of the PASC Interpretations 1003.2-1992 #31, 38, 49, 50, 51, 52, 55, 56,  
17555 57, 61, 62, 63, 64, 65, and 78.



17556 **NAME**

17557       expand — convert tabs to spaces

17558 **SYNOPSIS**17559 UP       expand [-t *tablist*][*file* ...]

17560

17561 **DESCRIPTION**

17562       The *expand* utility shall write files or the standard input to the standard output with <tab>  
 17563       characters replaced with one or more <space> characters needed to pad to the next tab stop. Any  
 17564       <backspace> characters shall be copied to the output and cause the column position count for  
 17565       tab stop calculations to be decremented; the column position count shall not be decremented  
 17566       below zero.

17567 **OPTIONS**

17568       The *expand* utility shall conform to the System Interface Definitions volume of  
 17569       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

17570       The following option shall be supported:

17571       -t *tablist*   Specify the tab stops. The application shall ensure that the argument *tablist*  
 17572       consists of a single positive decimal integer or multiple positive decimal integers,  
 17573       separated by <blank> characters or commas, in ascending order. If a single number  
 17574       is given, tabs shall be set *tablist* column positions apart instead of the default 8. If  
 17575       multiple numbers are given, the tabs shall be set at those specific column positions.

17576       The application shall ensure that each tab-stop position *N* is an integer value  
 17577       greater than zero, and the list is in strictly ascending order. This is taken to mean  
 17578       that, from the start of a line of output, tabbing to position *N* shall cause the next  
 17579       character output to be in the (*N*+1)th column position on that line.

17580       In the event of *expand* having to process a <tab> character at a position beyond the  
 17581       last of those specified in a multiple tab-stop list, the <tab> character shall be  
 17582       replaced by a single <space> character in the output.

17583 **OPERANDS**

17584       The following operand shall be supported:

17585       *file*           The path name of a text file to be used as input.

17586 **STDIN**

17587       See the INPUT FILES section.

17588 **INPUT FILES**

17589       Input files shall be text files.

17590 **ENVIRONMENT VARIABLES**

17591       The following environment variables shall affect the execution of *expand*:

17592       *LANG*           Provide a default value for the internationalization variables that are unset or null.  
 17593       If *LANG* is unset or null, the corresponding value from the implementation-  
 17594       dependent default locale shall be used. If any of the internationalization variables  
 17595       contains an invalid setting, the utility shall behave as if none of the variables had  
 17596       been defined.

17597       *LC\_ALL*          If set to a non-empty string value, override the values of all the other  
 17598       internationalization variables.

17599       *LC\_CTYPE*       Determine the locale for the interpretation of sequences of bytes of text data as  
 17600       characters (for example, single-byte as opposed to multi-byte characters in



17601 arguments and input files), the processing of <tab> and <space> characters, and  
 17602 for the determination of the width in column positions each character would  
 17603 occupy on an output device.

#### 17604 **LC\_MESSAGES**

17605 Determine the locale that should be used to affect the format and contents of  
 17606 diagnostic messages written to standard error.

17607 XSI **NLSPATH** Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

#### 17608 **ASYNCHRONOUS EVENTS**

17609 Default.

#### 17610 **STDOUT**

17611 The standard output shall be equivalent to the input files with <tab> characters converted into  
 17612 the appropriate number of <space> characters.

#### 17613 **STDERR**

17614 Used only for diagnostic messages.

#### 17615 **OUTPUT FILES**

17616 None.

#### 17617 **EXTENDED DESCRIPTION**

17618 None.

#### 17619 **EXIT STATUS**

17620 The following exit values shall be returned:

17621 0 Successful completion

17622 >0 An error occurred.

#### 17623 **CONSEQUENCES OF ERRORS**

17624 The *expand* utility shall terminate with an error message and non-zero exit status upon  
 17625 encountering difficulties accessing one of the *file* operands.

#### 17626 **APPLICATION USAGE**

17627 Application writers should note that this utility need not be provided on systems that do not  
 17628 support the User Portability Utilities option.

#### 17629 **EXAMPLES**

17630 None.

#### 17631 **RATIONALE**

17632 The *expand* utility is useful for preprocessing text files (before sorting, looking at specific  
 17633 columns, and so on) that contain <tab>s.

17634 See the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 3.106, Column  
 17635 Position.

17636 The *tablist* option-argument consists of integers in ascending order. Utility Syntax Guideline 8  
 17637 mandates that *expand* shall accept the integers (within the single argument) separated using  
 17638 either commas or <blank>s.

#### 17639 **FUTURE DIRECTIONS**

17640 None.

17641 **SEE ALSO**17642 *tabs, unexpand*17643 **CHANGE HISTORY**

17644 First released in Issue 4.

17645 **Issue 6**

17646 This utility is now marked as part of the User Portability Utilities option.

17647 The APPLICATION USAGE section is added.

17648 The obsolescent SYNOPSIS is removed.

17649 The *LC\_CTYPE* environment variable description is updated to align with the IEEE P1003.2b  
17650 draft standard.

17651 The normative text is reworded to avoid use of the term “must” for application requirements.

17652 **NAME**17653        *expr* — evaluate arguments as an expression17654 **SYNOPSIS**17655        *expr operand*17656 **DESCRIPTION**17657        The *expr* utility shall evaluate an expression and write the result to standard output.17658 **OPTIONS**

17659        None.

17660 **OPERANDS**

17661        The single expression evaluated by *expr* shall be formed from the operands, as described in the  
 17662        EXTENDED DESCRIPTION section. The application shall ensure that each of the expression  
 17663        operator symbols:

17664        ( ) | &amp; = &gt; &gt;= &lt; &lt;= != + - \* / % :

17665        and the symbols *integer* and *string* in the table are provided as separate arguments to *expr*.17666 **STDIN**

17667        Not used.

17668 **INPUT FILES**

17669        None.

17670 **ENVIRONMENT VARIABLES**17671        The following environment variables shall affect the execution of *expr*:

17672        *LANG*        Provide a default value for the internationalization variables that are unset or null.  
 17673        If *LANG* is unset or null, the corresponding value from the implementation-  
 17674        dependent default locale shall be used. If any of the internationalization variables  
 17675        contains an invalid setting, the utility shall behave as if none of the variables had  
 17676        been defined.

17677        *LC\_ALL*        If set to a non-empty string value, override the values of all the other  
 17678        internationalization variables.

17679        *LC\_COLLATE*    Determine the locale for the behavior of ranges, equivalence classes, and multi-  
 17680        character collating elements within regular expressions and by the string  
 17681        comparison operators.  
 17682

17683        *LC\_CTYPE*    Determine the locale for the interpretation of sequences of bytes of text data as  
 17684        characters (for example, single-byte as opposed to multi-byte characters in  
 17685        arguments) and the behavior of character classes within regular expressions.

17686        *LC\_MESSAGES*    Determine the locale that should be used to affect the format and contents of  
 17687        diagnostic messages written to standard error.  
 17688

17689 *XSI*        *NLSPATH*    Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

17690 **ASYNCHRONOUS EVENTS**

17691        Default.

17692 **STDOUT**

17693 The *expr* utility shall evaluate the expression and write the result, followed by a <newline>  
 17694 character, to standard output.

17695 **STDERR**

17696 Used only for diagnostic messages.

17697 **OUTPUT FILES**

17698 None.

17699 **EXTENDED DESCRIPTION**

17700 The formation of the expression to be evaluated is shown in the following table. The symbols  
 17701 *expr*, *expr1*, and *expr2* represent expressions formed from *integer* and *string* symbols and the  
 17702 expression operator symbols (all separate arguments) by recursive application of the constructs  
 17703 described in the table. The expressions are listed in order of increasing precedence, with equal-  
 17704 precedence operators grouped between horizontal lines. All of the operators shall be left-  
 17705 associative.

| Expression                                                                                                                                                                                | Description                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>expr1</i>   <i>expr2</i>                                                                                                                                                               | Returns the evaluation of <i>expr1</i> if it is neither null nor zero; otherwise, returns the evaluation of <i>expr2</i> if it is not null; otherwise, zero.                                                                                                                                                                                                                                               |
| <i>expr1</i> & <i>expr2</i>                                                                                                                                                               | Returns the evaluation of <i>expr1</i> if neither expression evaluates to null or zero; otherwise, returns zero.                                                                                                                                                                                                                                                                                           |
| <i>expr1</i> = <i>expr2</i><br><i>expr1</i> > <i>expr2</i><br><i>expr1</i> >= <i>expr2</i><br><i>expr1</i> < <i>expr2</i><br><i>expr1</i> <= <i>expr2</i><br><i>expr1</i> != <i>expr2</i> | Returns the result of a decimal integer comparison if both arguments are integers; otherwise, returns the result of a string comparison using the locale-specific collation sequence. The result of each comparison is 1 if the specified relationship is true, or 0 if the relationship is false.<br>Equal.<br>Greater than.<br>Greater than or equal.<br>Less than.<br>Less than or equal.<br>Not equal. |
| <i>expr1</i> + <i>expr2</i><br><i>expr1</i> − <i>expr2</i>                                                                                                                                | Addition of decimal integer-valued arguments.<br>Subtraction of decimal integer-valued arguments.                                                                                                                                                                                                                                                                                                          |
| <i>expr1</i> * <i>expr2</i><br><i>expr1</i> / <i>expr2</i><br><i>expr1</i> % <i>expr2</i>                                                                                                 | Multiplication of decimal integer-valued arguments.<br>Integer division of decimal integer-valued arguments, producing an integer result.<br>Remainder of integer division of decimal integer-valued arguments.                                                                                                                                                                                            |
| <i>expr1</i> : <i>expr2</i>                                                                                                                                                               | Matching expression; see below.                                                                                                                                                                                                                                                                                                                                                                            |
| ( <i>expr</i> )                                                                                                                                                                           | Grouping symbols. Any expression can be placed within parentheses. Parentheses can be nested to a depth of {EXPR_NEST_MAX}.                                                                                                                                                                                                                                                                                |
| <i>integer</i><br><i>string</i>                                                                                                                                                           | An argument consisting only of an (optional) unary minus followed by digits.<br>A string argument; see below.                                                                                                                                                                                                                                                                                              |

17737 **Matching Expression**

17738 The ' : ' matching operator shall compare the string resulting from the evaluation of *expr1* with  
 17739 the regular expression pattern resulting from the evaluation of *expr2*. Regular expression syntax  
 17740 shall be that defined in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section  
 17741 9.3, Basic Regular Expressions, except that all patterns are anchored to the beginning of the  
 17742 string (that is, only sequences starting at the first character of a string are matched by the regular  
 17743 expression) and, therefore, it is unspecified whether '^' is a special character in that context.  
 17744 Usually, the matching operator shall return a string representing the number of characters  
 17745 matched ('0' on failure). Alternatively, if the pattern contains at least one regular expression  
 17746 subexpression "[ \ ( . . . \ ) ]", the string corresponding to "\1" shall be returned.

17747 **String Operand**

17748 A string argument is an argument that cannot be identified as an *integer* argument or as one of  
 17749 the expression operator symbols shown in the OPERANDS section.

17750 The use of string arguments **length**, **substr**, **index**, or **match** produces unspecified results.

17751 **EXIT STATUS**

17752 The following exit values shall be returned:

- 17753 0 The *expression* evaluates to neither null nor zero.
- 17754 1 The *expression* evaluates to null or zero.
- 17755 2 Invalid *expression*.
- 17756 >2 An error occurred.

17757 **CONSEQUENCES OF ERRORS**

17758 Default.

17759 **APPLICATION USAGE**

17760 After argument processing by the shell, *expr* is not required to be able to tell the difference  
 17761 between an operator and an operand except by the value. If "\$a" is '=', the command:

17762 `expr $a = '='`

17763 looks like:

17764 `expr = = =`

17765 as the arguments are passed to *expr* (and they all may be taken as the '=' operator). The  
 17766 following works reliably:

17767 `expr X$a = X=`

17768 Also note that this volume of IEEE Std. 1003.1-200x permits implementations to extend utilities.  
 17769 The *expr* utility permits the integer arguments to be preceded with a unary minus. This means  
 17770 that an integer argument could look like an option. Therefore, the portable application must  
 17771 employ the "—" construct of Guideline 10 of the System Interface Definitions volume of  
 17772 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines to protect its operands if there is  
 17773 any chance the first operand might be a negative integer (or any string with a leading minus).

17774 **EXAMPLES**

17775 The *expr* utility has a rather difficult syntax:

- 17776 • Many of the operators are also shell control operators or reserved words, so they have to be  
 17777 escaped on the command line.

- Each part of the expression is composed of separate arguments, so liberal usage of <blank> characters is required. For example:

| Invalid                         | Valid                              |
|---------------------------------|------------------------------------|
| <code>expr 1+2</code>           | <code>expr 1 + 2</code>            |
| <code>expr "1 + 2"</code>       | <code>expr 1 + 2</code>            |
| <code>expr 1 + ( 2 * 3 )</code> | <code>expr 1 + \( 2 \* 3 \)</code> |

In many cases, the arithmetic and string features provided as part of the shell command language are easier to use than their equivalents in *expr*. Newly written scripts should avoid *expr* in favor of the new features within the shell; see Section 2.5 on page 43 and Section 2.6.4 on page 56.

The following command:

```
a=$(expr $a + 1)
```

adds 1 to the variable *a*.

The following command, for "*\$a*" equal to either */usr/abc/file* or just *file*:

```
expr $a : '.*\/(.*\)' \| $a
```

returns the last segment of a path name (that is, *file*). Applications should avoid the character *'/'* used alone as an argument: *expr* may interpret it as the division operator.

The following command:

```
expr "//$a" : '.*\/(.*\)'
```

is a better representation of the previous example. The addition of the *"/"* characters eliminates any ambiguity about the division operator and simplifies the whole expression. Also note that path names may contain characters contained in the *IFS* variable and should be quoted to avoid having "*\$a*" expand into multiple arguments.

The following command:

```
expr "$VAR" : '.*'
```

returns the number of characters in *VAR*.

#### RATIONALE

In an early proposal, EREs were used in the matching expression syntax. This was changed to BREs to avoid breaking historical applications.

The use of a leading circumflex in the BRE is unspecified because many historical implementations have treated it as a special character, despite their system documentation. For example:

```
expr foo : ^foo expr ^foo : ^foo
```

return 3 and 0, respectively, on those systems; their documentation would imply the reverse. Thus, the anchoring condition is left unspecified to avoid breaking historical scripts relying on this undocumented feature.

#### FUTURE DIRECTIONS

None.

17817 **SEE ALSO**

17818           Section 2.6.4

17819 **CHANGE HISTORY**

17820           First released in Issue 2.

17821 **Issue 4**

17822           Aligned with the ISO/IEC 9945-2: 1993 standard.

17823 **Issue 5**

17824           FUTURE DIRECTIONS section added.

17825 **Issue 6**17826           The *expr* utility is aligned with the IEEE P1003.2b draft standard, to include resolution of PASC

17827           Interpretation 1003.2-92 #104.

17828           The normative text is reworded to avoid use of the term “must” for application requirements.

17829 **NAME**

17830 false — return false value

17831 **SYNOPSIS**

17832 false

17833 **DESCRIPTION**

17834 The *false* utility shall return with a non-zero exit code.

17835 **OPTIONS**

17836 None.

17837 **OPERANDS**

17838 None.

17839 **STDIN**

17840 Not used.

17841 **INPUT FILES**

17842 None.

17843 **ENVIRONMENT VARIABLES**

17844 None.

17845 **ASYNCHRONOUS EVENTS**

17846 Default.

17847 **STDOUT**

17848 Not used.

17849 **STDERR**

17850 None.

17851 **OUTPUT FILES**

17852 None.

17853 **EXTENDED DESCRIPTION**

17854 None.

17855 **EXIT STATUS**

17856 The *false* utility always shall exit with a value other than zero.

17857 **CONSEQUENCES OF ERRORS**

17858 Default.

17859 **APPLICATION USAGE**

17860 None.

17861 **EXAMPLES**

17862 None.

17863 **RATIONALE**

17864 None.

17865 **FUTURE DIRECTIONS**

17866 None.

17867 **SEE ALSO**

17868 *true*



17869 **CHANGE HISTORY**

17870 First released in Issue 2.

17871 **Issue 4**

17872 Aligned with the ISO/IEC 9945-2: 1993 standard.

## 17873 NAME

17874 fc — process the command history list

## 17875 SYNOPSIS

17876 UP `fc [-r][-e editor] [first[last]]`17877 `fc -l[-nr] [first[last]]`17878 `fc -s[old=new][first]`

17879

## 17880 DESCRIPTION

17881 The *fc* utility shall list, or shall edit and re-execute, commands previously entered to an  
17882 interactive *sh*.

17883 The command history list shall reference commands by number. The first number in the list is  
17884 selected arbitrarily. The relationship of a number to its command shall not change except when  
17885 the user logs in and no other process is accessing the list, at which time the system may reset the  
17886 numbering to start the oldest retained command at another number (usually 1). When the  
17887 number reaches an implementation-dependent upper limit, which shall be no smaller than the  
17888 value in *HISTSIZE* or 32 767 (whichever is greater), the shell may wrap the numbers, starting the  
17889 next command with a lower number (usually 1). However, despite this optional wrapping of  
17890 numbers, *fc* shall maintain the time-ordering sequence of the commands. For example, if four  
17891 commands in sequence are given the numbers 32 766, 32 767, 1 (wrapped), and 2 as they are  
17892 executed, command 32 767 is considered the command previous to 1, even though its number is  
17893 higher.

17894 When commands are edited (when the *-l* option is not specified), the resulting lines shall be  
17895 entered at the end of the history list and then re-executed by *sh*. The *fc* command that caused the  
17896 editing shall not be entered into the history list. If the editor returns a non-zero exit status, this  
17897 shall suppress the entry into the history list and the command re-execution. Any command line  
17898 variable assignments or redirection operators used with *fc* shall affect both the *fc* command itself  
17899 as well as the command that results; for example:

17900 `fc -s -- -l 2>/dev/null`

17901 reinvoke the previous command, suppressing standard error for both *fc* and the previous  
17902 command.

## 17903 OPTIONS

17904 The *fc* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
17905 Section 12.2, Utility Syntax Guidelines.

17906 The following options shall be supported:

17907 **-e editor** Use the editor named by *editor* to edit the commands. The *editor* string is a utility  
17908 name, subject to search via the *PATH* variable (see the System Interface Definitions  
17909 volume of IEEE Std. 1003.1-200x, Chapter 8, Environment Variables). The value in  
17910 the *FCEDIT* variable shall be used as a default when *-e* is not specified. If *FCEDIT*  
17911 is null or unset, *ed* shall be used as the editor.

17912 **-l** (The letter ell.) List the commands rather than invoking an editor on them. The  
17913 commands shall be written in the sequence indicated by the *first* and *last* operands,  
17914 as affected by *-r*, with each command preceded by the command number.

17915 **-n** Suppress command numbers when listing with *-l*.

17916 **-r** Reverse the order of the commands listed (with *-l*) or edited (with neither *-l* nor  
17917 *-s*).

17918            **-s**            Reexecute the command without invoking an editor.

17919 **OPERANDS**

17920            The following operands shall be supported:

17921            *first, last*

17922                    Select the commands to list or edit. The number of previous commands that can be  
17923                    accessed shall be determined by the value of the *HISTSIZE* variable. The value of  
17924                    *first* or *last* or both shall be one of the following:

17925                    [+]*number*    A positive number representing a command number; command  
17926                    numbers can be displayed with the **-l** option.

17927                    -*number*       A negative decimal number representing the command that was  
17928                    executed *number* of commands previously. For example, -1 is the  
17929                    immediately previous command.

17930                    *string*        A string indicating the most recently entered command that begins  
17931                    with that string. If the *old=new* operand is not also specified with **-s**,  
17932                    the string form of the *first* operand cannot contain an embedded  
17933                    equal sign.

17934            When the synopsis form with **-s** is used:

17935                    

- If *first* is omitted, the previous command shall be used.

17936            For the synopsis forms without **-s**:

17937                    

- If *last* is omitted, *last* shall default to the previous command when **-l** is  
17938                    specified; otherwise, it shall default to *first*.

17939                    

- If *first* and *last* are both omitted, the previous 16 commands shall be listed or  
17940                    the previous single command shall be edited (based on the **-l** option).

17941                    

- If *first* and *last* are both present, all of the commands from *first* to *last* shall be  
17942                    edited (without **-l**) or listed (with **-l**). Editing multiple commands shall be  
17943                    accomplished by presenting to the editor all of the commands at one time, each  
17944                    command starting on a new line. If *first* represents a newer command than *last*,  
17945                    the commands shall be listed or edited in reverse sequence, equivalent to using  
17946                    **-r**. For example, the following commands on the first line are equivalent to the  
17947                    corresponding commands on the second:

17948                            fc -r 10 20       fc       30 40  
17949                            fc       20 10       fc -r 40 30

17950                    

- When a range of commands is used, it shall not be an error to specify *first* or *last*  
17951                    values that are not in the history list; *fc* shall substitute the value representing  
17952                    the oldest or newest command in the list, as appropriate. For example, if there  
17953                    are only ten commands in the history list, numbered 1 to 10:

17954                            fc -l  
17955                            fc 1 99

17956                    shall list and edit, respectively, all ten commands.

17957            *old=new*        Replace the first occurrence of string *old* in the commands to be re-executed by the  
17958                    string *new*.

17959 **STDIN**

17960 Not used.

17961 **INPUT FILES**

17962 None.

17963 **ENVIRONMENT VARIABLES**17964 The following environment variables shall affect the execution of *fc*:

17965 **FCEDIT** This variable, when expanded by the shell, shall determine the default value for  
 17966 the **-e** *editor* option's *editor* option-argument. If *FCEDIT* is null or unset, *ed* shall be  
 17967 used as the editor.

17968 **HISTFILE** Determine a path name naming a command history file. If the *HISTFILE* variable is  
 17969 not set, the shell may attempt to access or create a file **.sh\_history** in the directory  
 17970 referred to by the *HOME* environment variable. If the shell cannot obtain both read  
 17971 and write access to, or create, the history file, it shall use an unspecified  
 17972 mechanism that allows the history to operate properly. (References to history  
 17973 “file” in this section shall be understood to mean this unspecified mechanism in  
 17974 such cases.) An implementation may choose to access this variable only when  
 17975 initializing the history file; this initialization shall occur when *fc* or *sh* first attempt  
 17976 to retrieve entries from, or add entries to, the file, as the result of commands issued  
 17977 by the user, the file named by the *ENV* variable, or implementation-dependent  
 17978 system start-up files. (The initialization process for the history file can be  
 17979 dependent on the system start-up files, in that they may contain commands that  
 17980 effectively preempt the user's settings of *HISTFILE* and *HISTSIZE*. For example,  
 17981 function definition commands are recorded in the history file, unless the **set -o**  
 17982 **nolog** option is set. If the system administrator includes function definitions in  
 17983 some system start-up file called before the *ENV* file, the history file is initialized  
 17984 before the user gets a chance to influence its characteristics.) In some historical  
 17985 shells, the history file is initialized just after the *ENV* file has been processed.  
 17986 Therefore, it is implementation-dependent whether changes made to *HISTFILE*  
 17987 after the history file has been initialized are effective. Implementations may  
 17988 choose to disable the history list mechanism for users with appropriate privileges  
 17989 who do not set *HISTFILE*; the specific circumstances under which this occurs are  
 17990 implementation-dependent. If more than one instance of the shell is using the  
 17991 same history file, it is unspecified how updates to the history file from those shells  
 17992 interact. As entries are deleted from the history file, they shall be deleted oldest  
 17993 first. It is unspecified when history file entries are physically removed from the  
 17994 history file.

17995 **HISTSIZE** Determine a decimal number representing the limit to the number of previous  
 17996 commands that are accessible. If this variable is unset, an unspecified default  
 17997 greater than or equal to 128 shall be used. The maximum number of commands in  
 17998 the history list is unspecified, but shall be at least 128. An implementation may  
 17999 choose to access this variable only when initializing the history file, as described  
 18000 under *HISTFILE*. Therefore, it is unspecified whether changes made to *HISTSIZE*  
 18001 after the history file has been initialized are effective.

18002 **LANG** Provide a default value for the internationalization variables that are unset or null.  
 18003 If *LANG* is unset or null, the corresponding value from the implementation-  
 18004 dependent default locale shall be used. If any of the internationalization variables  
 18005 contains an invalid setting, the utility shall behave as if none of the variables had  
 18006 been defined.

18007 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 18008 internationalization variables.

18009 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 18010 characters (for example, single-byte as opposed to multi-byte characters in  
 18011 arguments and input files).

18012 **LC\_MESSAGES**  
 18013 Determine the locale that should be used to affect the format and contents of  
 18014 diagnostic messages written to standard error.

18015 XSI **NLSPATH** Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

18016 **ASYNCHRONOUS EVENTS**  
 18017 Default.

18018 **STDOUT**  
 18019 When the **-l** option is used to list commands, the format of each command in the list shall be as  
 18020 follows:  
 18021 "%d\t%s\n", <line number>, <command>  
 18022 If both the **-l** and **-n** options are specified, the format of each command shall be:  
 18023 "\t%s\n", <command>  
 18024 If the <command> consists of more than one line, the lines after the first shall be displayed as:  
 18025 "\t%s\n", <continued-command>

18026 **STDERR**  
 18027 Used only for diagnostic messages.

18028 **OUTPUT FILES**  
 18029 None.

18030 **EXTENDED DESCRIPTION**  
 18031 None.

18032 **EXIT STATUS**  
 18033 The following exit values shall be returned:  
 18034 0 Successful completion of the listing.  
 18035 >0 An error occurred.  
 18036 Otherwise, the exit status shall be that of the commands executed by *fc*.

18037 **CONSEQUENCES OF ERRORS**  
 18038 Default.

18039 **APPLICATION USAGE**  
 18040 Since editors sometimes use file descriptors as integral parts of their editing, redirecting their file  
 18041 descriptors as part of the *fc* command can produce unexpected results. For example, if *vi* is the  
 18042 *FCEDIT* editor, the command:  
 18043 *fc -s | more*  
 18044 does not work correctly on many systems.  
 18045 Users on windowing systems may want to have separate history files for each window by  
 18046 setting *HISTFILE* as follows:

18047 HISTFILE=\$HOME/.sh\_hist\$\$

18048 Application writers should note that this utility need not be provided on systems that do not  
18049 support the User Portability Utilities option.

#### 18050 EXAMPLES

18051 None.

#### 18052 RATIONALE

18053 This utility is based on the *fc* built-in of the KornShell.

18054 An early proposal specified the *-e* option as [*-e editor [old= new ]*], which is not historical  
18055 practice. Historical practice in *fc* of either [*-e editor*] or [*-e - [old= new ]*] is acceptable, but not  
18056 both together. To clarify this, a new option *-s* was introduced replacing the [*-e -*]. This resolves  
18057 the conflict and makes *fc* conform to the Utility Syntax Guidelines.

18058 *HISTFILE* Users on windowing systems may want to have separate history files for each  
18059 window by setting *HISTFILE* as follows:

18060 HISTFILE=\$HOME/.sh\_hist\$\$

18061 Some implementations of the KornShell check for the superuser and do not create  
18062 a history file unless *HISTFILE* is set. This is done primarily to avoid creating  
18063 unlinked files in the root file system when logging in during single-user mode.  
18064 *HISTFILE* must be set for the superuser to have history.

18065 *HISTSIZE* Needed to limit the size of history files. It is the intent of the standard developers  
18066 that when two shells share the same history file, commands that are entered in one  
18067 shell shall be accessible by the other shell. Because of the difficulties of  
18068 synchronization over a network, the exact nature of the interaction is unspecified.

18069 The initialization process for the history file can be dependent on the system start-up files, in  
18070 that they may contain commands that effectively preempt the settings the user has for *HISTFILE*  
18071 and *HISTSIZE*. For example, function definition commands are recorded in the history file. If the  
18072 system administrator includes function definitions in some system start-up file called before the  
18073 *ENV* file, the history file is initialized before the user can influence its characteristics. In some  
18074 historical shells, the history file is initialized just after the *ENV* file has been processed. Because  
18075 of these situations, the text requires the initialization process to be implementation-dependent.

18076 Consideration was given to omitting the *fc* utility in favor of the command line editing feature in  
18077 *sh*. For example, in *vi* editing mode, typing "<ESC> v" is equivalent to:

18078 EDITOR=vi fc

18079 However, the *fc* utility allows the user the flexibility to edit multiple commands simultaneously  
18080 (such as *fc 10 20*) and to use editors other than those supported by *sh* for command line editing.

18081 In the KornShell, the alias *r* ("re-do") is preset to *fc -e -* (equivalent to the POSIX *fc -s*). This is  
18082 probably an easier command name to remember than *fc* ("fix command"), but it does not meet  
18083 the Utility Syntax Guidelines. Renaming *fc* to *hist* or *redo* was considered, but since this  
18084 description closely matches historical KornShell practice already, such a renaming was seen as  
18085 gratuitous. Users are free to create aliases whenever odd historical names such as *fc*, *awk*, *cat*,  
18086 *grep*, or *yacc* are standardized by POSIX.

18087 Command numbers have no ordering effects; they are like serial numbers. The *-r* option and  
18088 *-number* operand address the sequence of command execution, regardless of serial numbers. So,  
18089 for example, if the command number wrapped back to 1 at some arbitrary point, there would be  
18090 no ambiguity associated with traversing the wrap point. For example, if the command history  
18091 were:

18092            32766: echo 1  
18093            32767: echo 2  
18094            1: echo 3  
  
18095            the number -2 refers to command 32 767 because it is the second previous command, regardless  
18096            of serial number.  
  
18097 **FUTURE DIRECTIONS**  
18098            None.  
  
18099 **SEE ALSO**  
18100            *sh*  
  
18101 **CHANGE HISTORY**  
18102            First released in Issue 4.  
  
18103 **Issue 5**  
18104            FUTURE DIRECTIONS section added.  
  
18105 **Issue 6**  
18106            This utility is now marked as part of the User Portability Utilities option.  
  
18107            In the ENVIRONMENT VARIABLES section, the text “user’s home directory” is updated to  
18108            “directory referred to by the *HOME* environment variable”.

18109 **NAME**

18110 fg — run jobs in the foreground

18111 **SYNOPSIS**18112 UP fg [*job\_id*]

18113

18114 **DESCRIPTION**

18115 If job control is enabled (see the description of *set -m*), the *fg* utility shall move a background job  
 18116 from the current environment (see Section 2.12 on page 90) into the foreground.

18117 Using *fg* to place a job into the foreground shall remove its process ID from the list of those  
 18118 “known in the current shell execution environment”; see Section 2.9.3.1 on page 74.

18119 **OPTIONS**

18120 None.

18121 **OPERANDS**

18122 The following operand shall be supported:

18123 *job\_id* Specify the job to be run as a foreground job. If no *job\_id* operand is given, the  
 18124 *job\_id* for the job that was most recently suspended, placed in the background or  
 18125 run as a background job, shall be used. The format of *job\_id* is described in the  
 18126 System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 3.207, Job  
 18127 Control Job ID.

18128 **STDIN**

18129 Not used.

18130 **INPUT FILES**

18131 None.

18132 **ENVIRONMENT VARIABLES**18133 The following environment variables shall affect the execution of *fg*:

18134 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 18135 If *LANG* is unset or null, the corresponding value from the implementation-  
 18136 dependent default locale shall be used. If any of the internationalization variables  
 18137 contains an invalid setting, the utility shall behave as if none of the variables had  
 18138 been defined.

18139 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 18140 internationalization variables.

18141 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 18142 characters (for example, single-byte as opposed to multi-byte characters in  
 18143 arguments).

18144 *LC\_MESSAGES*

18145 Determine the locale that should be used to affect the format and contents of  
 18146 diagnostic messages written to standard error.

18147 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.18148 **ASYNCHRONOUS EVENTS**

18149 Default.



**18150 STDOUT**

18151 The *fg* utility shall write the command line of the job to standard output in the following format:

18152 "%s\n", <command>

**18153 STDERR**

18154 Used only for diagnostic messages.

**18155 OUTPUT FILES**

18156 None.

**18157 EXTENDED DESCRIPTION**

18158 None.

**18159 EXIT STATUS**

18160 The following exit values shall be returned:

18161 0 Successful completion.

18162 >0 An error occurred.

**18163 CONSEQUENCES OF ERRORS**

18164 If job control is disabled, the *fg* utility shall exit with an error and no job shall be placed in the foreground.

**18166 APPLICATION USAGE**

18167 The *fg* utility does not work as expected when it is operating in its own utility execution environment because that environment has no applicable jobs to manipulate. See the APPLICATION USAGE section for *bg* on page 243. For this reason, *fg* is generally implemented as a shell regular built-in.

18171 Application writers should note that this utility need not be provided on systems that do not support the User Portability Utilities option.

**18173 EXAMPLES**

18174 None.

**18175 RATIONALE**

18176 The extensions to the shell specified in this volume of IEEE Std. 1003.1-200x have mostly been based on features provided by the KornShell. The job control features provided by *bg*, *fg*, and *jobs* are also based on the KornShell. The standard developers examined the characteristics of the C shell versions of these utilities and found that differences exist. Despite widespread use of the C shell, the KornShell versions were selected for this volume of IEEE Std. 1003.1-200x to maintain a degree of uniformity with the rest of the KornShell features selected (such as the very popular command line editing features).

**18183 FUTURE DIRECTIONS**

18184 None.

**18185 SEE ALSO**

18186 *bg*, *kill*, *jobs*, *wait*

**18187 CHANGE HISTORY**

18188 First released in Issue 4.

**18189 Issue 6**

18190 This utility is now marked as part of the User Portability Utilities option.

18191 The APPLICATION USAGE section is added.

 18192

The JC marking is removed from the SYNOPSIS since job control is mandatory is this issue.



18193 **NAME**

18194       file — determine file type

18195 **SYNOPSIS**

18196 UP       file [-dhi][-M file][-m file] file ...

18197

18198 **DESCRIPTION**18199       The *file* utility shall perform a series of tests on each specified *file* in an attempt to classify it:18200           1. If the file is not a regular file, its file type shall be identified. The file types directory, FIFO,  
18201           block special, and character special shall be identified as such. Other implementation-  
18202           dependent file types may also be identified.

18203           2. If the file is a regular file, and:

18204               a. The file is zero-length, it shall be identified as an empty file.

18205               b. The file is not zero-length, *file* shall examine an initial segment of the file and shall  
18206               make a guess at identifying its contents or whether it is an executable binary file.  
18207               (The answer is not guaranteed to be correct.)18208       If *file* does not exist, cannot be read, or its file status could not be determined, the output shall  
18209       indicate that the file was processed, but that its type could not be determined.18210       If *file* is a symbolic link, by default the link shall be resolved and *file* shall test the type of file  
18211       referenced by the symbolic link.18212 **OPTIONS**18213       The *file* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
18214       Section 12.2, Utility Syntax Guidelines.

18215       The following options shall be supported by the implementation:

18216       **-d**           Apply any default system tests to the file.18217       **-h**           When a symbolic link is encountered, identify the file as a symbolic link. If **-h** is  
18218       not specified and *file* is a symbolic link that refers to a nonexistent file, *file* shall  
18219       identify the file as a symbolic link, as if **-h** had been specified.18220       **-i**           If a file is a regular file, do not attempt to classify the type of the file further, but  
18221       identify the file as specified in the STDOUT section, using a <type> string that  
18222       contains the string regular file.18223       **-M file**       Specify the name of a file containing tests that shall be applied to a file in order to  
18224       classify it (see the EXTENDED DESCRIPTION). No default system tests shall be  
18225       applied.18226       **-m file**       Specify the name of a file containing tests that shall be applied to a file in order to  
18227       classify it (see the EXTENDED DESCRIPTION).18228       If multiple instances of the **-m**, **-d**, or **-M** options are specified, the concatenation of the tests  
18229       specified, in the order specified, shall be the set of tests that are applied. If a **-M** option is  
18230       specified, no tests other than those specified using the **-d**, **-M**, and **-m** options shall be applied  
18231       to the file. If neither the **-d** nor **-M** options are specified, any default system tests shall be  
18232       applied after any tests specified using the **-m** option.

18233 **OPERANDS**

18234 The following operand shall be supported:

18235 *file* A path name of a file to be tested.18236 **STDIN**

18237 Not used.

18238 **INPUT FILES**18239 The *file* can be any file type.18240 **ENVIRONMENT VARIABLES**18241 The following environment variables shall affect the execution of *file*:

18242 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 18243 If *LANG* is unset or null, the corresponding value from the implementation-  
 18244 dependent default locale shall be used. If any of the internationalization variables  
 18245 contains an invalid setting, the utility shall behave as if none of the variables had  
 18246 been defined.

18247 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 18248 internationalization variables.

18249 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 18250 characters (for example, single-byte as opposed to multi-byte characters in  
 18251 arguments and input files).

18252 *LC\_MESSAGES*

18253 Determine the locale that should be used to affect the format and contents of  
 18254 diagnostic messages written to standard error and informative messages written to  
 18255 standard output.

18256 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

18257 **ASYNCHRONOUS EVENTS**

18258 Default.

18259 **STDOUT**18260 In the POSIX locale, the following format shall be used to identify each operand, *file* specified:18261 "%s: %s\n", <*file*>, <*type*>

18262 The values for <*type*> are unspecified, except that in the POSIX locale, if *file* is identified as one  
 18263 of the types listed in the following table, <*type*> shall contain (but is not limited to) the  
 18264 corresponding string. Each space shown in the strings shall be exactly one <space> character.

Table 4-8 File Utility Output Strings

| If file is a:                                                | <type> shall contain the string: |
|--------------------------------------------------------------|----------------------------------|
| Directory                                                    | directory                        |
| FIFO                                                         | fifo                             |
| Block special                                                | block special                    |
| Character special                                            | character special                |
| Executable binary                                            | executable                       |
| Empty regular file                                           | empty                            |
| Symbolic link                                                | symbolic link to                 |
| <i>ar</i> archive library (see <i>ar</i> )                   | archive                          |
| Extended <i>cpio</i> format (see <i>pax</i> )                | cpio archive                     |
| Extended <i>tar</i> format (see <b>ustar</b> in <i>pax</i> ) | tar archive                      |
| Shell script                                                 | commands text                    |
| C-language source                                            | c program text                   |
| FORTRAN source                                               | fortran program text             |

If *file* is identified as a symbolic link (see **-h**), the following alternative output format shall be used:

```
"%s: %s %s\n", <file>, <type>, <contents of link>"
```

If the file named by the *file* operand does not exist or cannot be read, the string "cannot open" shall be included as part of the <type> field, but this shall not be considered an error that affects the exit status. If the type of the file named by the *file* operand cannot be determined, the string "data" shall be included as part of the <type> field, but this shall not be considered an error that affects the exit status.

## STDERR

Used only for diagnostic messages.

## OUTPUT FILES

None.

## EXTENDED DESCRIPTION

A file specified as an option-argument to the **-m** or **-M** options shall contain one test per line, which shall be applied to the file. If the test succeeds, the message field of the line shall be printed and no further tests shall be applied, with the exception that tests on immediately following lines beginning with a single '**>**' character shall be applied.

Each line shall be composed of the following four <blank>-separated fields:

**offset**      An unsigned number (optionally preceded by a single '**>**' character) specifying the *offset*, in bytes, of the value in the file that is to be compared against the *value* field of the line. If the file is shorter than the specified offset, the test shall fail.

If the *offset* begins with the character '**>**', the test contained in the line shall not be applied to the file unless the test on the last line for which the *offset* did not begin with a '**>**' was successful. By default, the *offset* shall be interpreted as an unsigned decimal number. With a leading 0x or 0X, the *offset* shall be interpreted as a hexadecimal number; otherwise, with a leading 0, the *offset* shall be interpreted as an octal number.

**type**      The type of the value in the file to be tested. The type shall consist of the type specification characters **c**, **d**, **f**, **s**, and **u**, specifying character, signed decimal, floating point, string, and unsigned decimal, respectively.

|       |                   |                                                                                                                                   |
|-------|-------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| 18310 |                   | The <i>type</i> string shall be interpreted as the bytes from the file starting at the                                            |
| 18311 |                   | specified <i>offset</i> and including the same number of bytes specified by the <i>value</i> field.                               |
| 18312 |                   | If insufficient bytes remain in the file past the <i>offset</i> to match the <i>value</i> field, the test                         |
| 18313 |                   | shall fail.                                                                                                                       |
| 18314 |                   | The type specification characters <b>d</b> , <b>f</b> , and <b>u</b> can be followed by an optional                               |
| 18315 |                   | unsigned decimal integer that specifies the number of bytes represented by the                                                    |
| 18316 |                   | type. The type specification character <b>f</b> can be followed by an optional <b>F</b> , <b>D</b> , or <b>L</b> ,                |
| 18317 |                   | indicating that the value is of type <b>float</b> , <b>double</b> , or <b>long double</b> , respectively. The                     |
| 18318 |                   | type specification characters <b>d</b> and <b>u</b> can be followed by an optional <b>C</b> , <b>S</b> , <b>I</b> , or <b>L</b> , |
| 18319 |                   | indicating that the value is of type <b>char</b> , <b>short</b> , <b>int</b> , or <b>long</b> , respectively.                     |
| 18320 |                   | The default number of bytes represented by the type specifiers <b>d</b> , <b>f</b> , and <b>u</b> shall                           |
| 18321 |                   | correspond to their respective C-language types as follows. If the system claims                                                  |
| 18322 |                   | conformance to the C-Language Development Utilities option, those specifiers                                                      |
| 18323 |                   | shall correspond to the default sizes used in the <i>c89</i> utility. Otherwise, the default                                      |
| 18324 |                   | sizes shall be implementation-dependent.                                                                                          |
| 18325 |                   | For the type specifier characters <b>d</b> and <b>u</b> , the default number of bytes shall                                       |
| 18326 |                   | correspond to the size of the basic integral data type of the implementation. For                                                 |
| 18327 |                   | these specifier characters, the implementation shall support values of the optional                                               |
| 18328 |                   | number of bytes to be converted corresponding to the number of bytes in the C-                                                    |
| 18329 |                   | language types <b>char</b> , <b>short</b> , <b>int</b> , or <b>long</b> . These numbers can also be specified by an               |
| 18330 |                   | application as the characters <b>C</b> , <b>S</b> , <b>I</b> , and <b>L</b> , respectively. The byte order used when              |
| 18331 |                   | interpreting numeric values is implementation-dependent, but shall correspond to                                                  |
| 18332 |                   | the order in which a constant of the corresponding type is stored in memory on the                                                |
| 18333 |                   | system.                                                                                                                           |
| 18334 |                   | For the type specifier <b>f</b> , the default number of bytes shall correspond to the number                                      |
| 18335 |                   | of bytes in the basic double precision floating-point data type of the underlying                                                 |
| 18336 |                   | implementation. The implementation shall support values of the optional number                                                    |
| 18337 |                   | of bytes to be converted corresponding to the number of bytes in the C-language                                                   |
| 18338 |                   | types <b>float</b> , <b>double</b> , and <b>long double</b> . These numbers can also be specified by an                           |
| 18339 |                   | application as the characters <b>F</b> , <b>D</b> , and <b>L</b> , respectively.                                                  |
| 18340 |                   | All type specifiers, except for <b>s</b> , can be followed by a mask specifier of the form                                        |
| 18341 |                   | <b>&amp;number</b> . The mask value shall be AND'ed with the value before the comparison                                          |
| 18342 |                   | with the value from the file is made. By default, the mask shall be interpreted as an                                             |
| 18343 |                   | unsigned decimal number. With a leading 0x or 0X, the mask shall be interpreted                                                   |
| 18344 |                   | as an unsigned hexadecimal number; otherwise, with a leading 0, the mask shall be                                                 |
| 18345 |                   | interpreted as an unsigned octal number.                                                                                          |
| 18346 |                   | The strings <b>byte</b> , <b>short</b> , <b>long</b> , and <b>string</b> shall also be supported as type fields,                  |
| 18347 |                   | being interpreted as <b>dC</b> , <b>dS</b> , <b>dL</b> , and <b>s</b> , respectively.                                             |
| 18348 | <i>value</i>      | The <i>value</i> to be compared with the value from the file.                                                                     |
| 18349 |                   | Any value that contains a character that is not a digit, other than a leading sign                                                |
| 18350 |                   | ('+' or '-') or a leading 0x or 0X, shall be interpreted as a string. The test shall                                              |
| 18351 |                   | succeed only when a string value exactly matches the bytes from the file.                                                         |
| 18352 |                   | If the <i>value</i> is a string, it can contain the following sequences:                                                          |
| 18353 | <i>\character</i> | The backslash-escape sequences as specified in the System                                                                         |
| 18354 |                   | Interface Definitions volume of IEEE Std. 1003.1-200x, Table 5-1,                                                                 |
| 18355 |                   | Escape Sequences and Associated Actions ('\\', '\a', '\b',                                                                        |
| 18356 |                   | '\f', '\n', '\r', '\t', '\v'). The results of using any other                                                                     |

|       |                               |                                                                                                            |
|-------|-------------------------------|------------------------------------------------------------------------------------------------------------|
| 18357 |                               | character, other than an octal digit, following the backslash are                                          |
| 18358 |                               | unspecified.                                                                                               |
| 18359 | <i>\octal</i>                 | Octal sequences that can be used to represent characters with                                              |
| 18360 |                               | specific coded values. An octal sequence shall consist of a                                                |
| 18361 |                               | backslash followed by the longest sequence of one, two, or three                                           |
| 18362 |                               | octal-digit characters (01234567). If the size of a byte on the                                            |
| 18363 |                               | system is greater than 9 bits, the valid escape sequence used to                                           |
| 18364 |                               | represent a byte is implementation-dependent.                                                              |
| 18365 |                               | By default, any value that is not a string shall be interpreted as a signed decimal                        |
| 18366 |                               | number. Any such value, with a leading 0x or 0X, shall be interpreted as an                                |
| 18367 |                               | unsigned hexadecimal number; otherwise, with a leading zero, the value shall be                            |
| 18368 |                               | interpreted as an unsigned octal number.                                                                   |
| 18369 |                               | If the value is not a string, it can be preceded by a character indicating the                             |
| 18370 |                               | comparison to be performed. Permissible characters and the comparisons they                                |
| 18371 |                               | specify are as follows:                                                                                    |
| 18372 | =                             | The test shall succeed if the value from the file equals the <i>value</i> field.                           |
| 18373 | <                             | The test shall succeed if the value from the file is less than the <i>value</i> field.                     |
| 18374 | >                             | The test shall succeed if the value from the file is greater than the <i>value</i> field.                  |
| 18375 | &                             | The test shall succeed if all of the bits in the <i>value</i> field are set in the value                   |
| 18376 |                               | from the file.                                                                                             |
| 18377 | ~                             | The test shall succeed if at least one of the bits in the <i>value</i> field is not set in the             |
| 18378 |                               | value from the file.                                                                                       |
| 18379 | x                             | The test shall succeed if there is any value in the file.                                                  |
| 18380 | <i>message</i>                | The <i>message</i> to be printed if the test succeeds. The <i>message</i> shall be interpreted             |
| 18381 |                               | using the notation for the <i>printf</i> formatting specification; see <i>printf</i> . If the <i>value</i> |
| 18382 |                               | field was a string, then the value from the file shall be the argument for the <i>printf</i>               |
| 18383 |                               | formatting specification; otherwise, the value from the file shall be the argument.                        |
| 18384 | <b>EXIT STATUS</b>            |                                                                                                            |
| 18385 |                               | The following exit values shall be returned:                                                               |
| 18386 | 0                             | Successful completion.                                                                                     |
| 18387 | >0                            | An error occurred.                                                                                         |
| 18388 | <b>CONSEQUENCES OF ERRORS</b> |                                                                                                            |
| 18389 |                               | Default.                                                                                                   |
| 18390 | <b>APPLICATION USAGE</b>      |                                                                                                            |
| 18391 |                               | The <i>file</i> utility can only be required to guess at many of the file types because only exhaustive    |
| 18392 |                               | testing can determine some types with certainty. For example, binary data on some systems                  |
| 18393 |                               | might match the initial segment of an executable or a <i>tar</i> archive.                                  |
| 18394 |                               | Note that the table indicates that the output contains the stated string. Systems may add text             |
| 18395 |                               | before or after the string. For executables, as an example, the machine architecture and various           |
| 18396 |                               | facts about how the file was link-edited may be included.                                                  |
| 18397 |                               | Application writers should note that this utility need not be provided on systems that do not              |
| 18398 |                               | support the User Portability Utilities option.                                                             |

18399 **EXAMPLES**

18400 Determine whether an argument is a binary executable file:

```
18401 file "$1" | grep -Fq executable &&
18402 printf "%s is executable.\n" "$1"
```

18403 **RATIONALE**

18404 The **-f** option was omitted because the same effect can (and should) be obtained using the *xargs*  
 18405 utility.

18406 Historical versions of the *file* utility attempt to identify the following types of files: symbolic link,  
 18407 directory, character special, block special, socket, *tar* archive, *cpio* archive, SCCS archive, archive  
 18408 library, empty, *compress* output, *pack* output, binary data, C source, FORTRAN source, assembler  
 18409 source, *nroff*/*troff*/*eqn*/*tbl* source *troff* output, shell script, C shell script, English text, ASCII text,  
 18410 various executables, APL workspace, compiled terminfo entries, and CURSES screen images.  
 18411 Only those types that are reasonably well specified in POSIX or are directly related to POSIX  
 18412 utilities are listed in the table.

18413 Implementations that support symbolic links are encouraged to use the string "symbolic  
 18414 link" to identify them.

18415 Historical systems have used a "magic file" named **/etc/magic** to help identify file types. Because  
 18416 it is generally useful for users and scripts to be able to identify special file types, the **-m** flag and  
 18417 a portable format for user-created magic files has been specified. No requirement is made that an  
 18418 implementation of *file* use this method of identifying files, only that users be permitted to add  
 18419 their own classifying tests.

18420 In addition, three options have been added to historical practice. The **-d** flag has been added to  
 18421 permit users to cause their tests to follow any default system tests. The **-i** flag has been added to  
 18422 permit users to test portably for regular files in shell scripts. The **-M** flag has been added to  
 18423 permit users to ignore any default system tests.

18424 The historical **-c** option was omitted as not particularly useful to users or portable shell scripts.  
 18425 In addition, a reasonable implementation of the *file* utility would report any errors found each  
 18426 time the magic file is read.

18427 The historical format of the magic file was the same as that specified by the Rationale in the  
 18428 previous version of IEEE Std. 1003.1-200x for the *offset*, *value*, and *message* fields; however, it  
 18429 used less precise type fields than the format specified by the current normative text. The new  
 18430 type field values are a superset of the historical ones.

18431 The following is an example magic file:

```
18432 0 short 070707 cpio archive
18433 0 short 0143561 Byte-swapped cpio archive
18434 0 string 070707 ASCII cpio archive
18435 0 long 0177555 Very old archive
18436 0 short 0177545 Old archive
18437 0 short 017437 Old packed data
18438 0 string \037\036 Packed data
18439 0 string \377\037 Compacted data
18440 0 string \037\235 Compressed data
18441 >2 byte&0x80 >0 Block compressed
18442 >2 byte&0x1f x %d bits
18443 0 string \032\001 Compiled Terminfo Entry
18444 0 short 0433 Curses screen image
18445 0 short 0434 Curses screen image
```



|       |   |        |                    |                                    |
|-------|---|--------|--------------------|------------------------------------|
| 18446 | 0 | string | <ar>               | System V Release 1 archive         |
| 18447 | 0 | string | !<arch>\n__.SYMDEF | Archive random library             |
| 18448 | 0 | string | !<arch>            | Archive                            |
| 18449 | 0 | string | ARF_BEGARF         | PHIGS clear text archive           |
| 18450 | 0 | long   | 0x137A2950         | Scalable OpenFont binary           |
| 18451 | 0 | long   | 0x137A2951         | Encrypted scalable OpenFont binary |

#### 18452 **FUTURE DIRECTIONS**

18453       None.

#### 18454 **SEE ALSO**

18455       *ls*

#### 18456 **CHANGE HISTORY**

18457       First released in Issue 4.

#### 18458 **Issue 6**

18459       This utility is now marked as part of the User Portability Utilities option. |

18460       Options and an EXTENDED DESCRIPTION are added as specified in the IEEE P1003.2b draft |  
 18461       standard. |

18462 **NAME**

18463       find — find files

18464 **SYNOPSIS**18465       find [-H | -L] *path* ... [*operand\_expression* ...]18466 **DESCRIPTION**

18467       The *find* utility shall recursively descend the directory hierarchy from each file specified by *path*,  
 18468       evaluating a Boolean expression composed of the primaries described in the OPERANDS section  
 18469       for each file encountered.

18470       The *find* utility shall be able to descend to arbitrary depths in a file hierarchy and shall not fail  
 18471       due to path length limitations (unless a *path* operand specified by the application exceeds  
 18472       {PATH\_MAX} requirements).

18473       The *find* utility shall detect infinite loops; that is, entering a previously visited directory that is an  
 18474       ancestor of the last file encountered. When it detects an infinite loop, *find* shall write a  
 18475       diagnostic message to standard error and shall either recover its position in the hierarchy or  
 18476       terminate.

18477 **OPTIONS**

18478       The *find* utility shall conform to the System Interface Definitions volume of  
 18479       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

18480       The following options shall be supported by the implementation:

18481       **-H**       Cause the file information and file type evaluated for each symbolic link  
 18482       encountered on the command line to be those of the file referenced by the link, and  
 18483       not the link itself. If the referenced file does not exist, the file information and type  
 18484       shall be for the link itself. File information for all symbolic links not on the  
 18485       command line shall be that of the link itself.

18486       **-L**       Cause the file information and file type evaluated for each symbolic link to be  
 18487       those of the file referenced by the link, and not the link itself.

18488       Specifying more than one of the mutually-exclusive options **-H** and **-L** shall not be considered  
 18489       an error. The last option specified shall determine the behavior of the utility.

18490 **OPERANDS**

18491       The following operands shall be supported:

18492       The *path* operand is a path name of a starting point in the directory hierarchy.

18493       The first argument that starts with a '-', or is a '!' or a '(', and all subsequent arguments  
 18494       shall be interpreted as an *expression* made up of the following primaries and operators. In the  
 18495       descriptions, wherever *n* is used as a primary argument, it shall be interpreted as a decimal  
 18496       integer optionally preceded by a plus ('+') or minus ('-') sign, as follows:

18497       +*n*   More than *n*.

18498       *n*     Exactly *n*.

18499       -*n*   Less than *n*.

18500       The following primaries shall be supported:

18501       **-name** *pattern*

18502       The primary shall evaluate as true if the basename of the file name being examined  
 18503       matches *pattern* using the pattern matching notation described in Section 2.13 on  
 18504       page 92.

|       |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------|----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 18505 | <b>-nouser</b>       | The primary shall evaluate as true if the file belongs to a user ID for which the <i>getpwuid()</i> function defined in the System Interfaces volume of IEEE Std. 1003.1-200x (or equivalent) returns NULL.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 18506 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18507 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18508 | <b>-nogroup</b>      | The primary shall evaluate as true if the file belongs to a group ID for which the <i>getgrgid()</i> function defined in the System Interfaces volume of IEEE Std. 1003.1-200x (or equivalent) returns NULL.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 18509 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18510 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18511 | <b>-xdev</b>         | The primary always shall evaluate as true; it shall cause <i>find</i> not to continue descending past directories that have a different device ID ( <i>st_dev</i> , see the <i>stat()</i> function defined in the System Interfaces volume of IEEE Std. 1003.1-200x). If any <b>-xdev</b> primary is specified, it shall apply to the entire expression even if the <b>-xdev</b> primary would not normally be evaluated.                                                                                                                                                                                                                                                                                                                                                        |
| 18512 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18513 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18514 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18515 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18516 | <b>-prune</b>        | The primary always shall evaluate as true; it shall cause <i>find</i> not to descend the current path name if it is a directory. If the <b>-depth</b> primary is specified, the <b>-prune</b> primary shall have no effect.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 18517 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18518 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18519 | <b>-perm [-]mode</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18520 |                      | The <i>mode</i> argument is used to represent file mode bits. It shall be identical in format to the <i>symbolic_mode</i> operand described in <i>chmod</i> on page 273, and shall be interpreted as follows. To start, a template shall be assumed with all file mode bits cleared. An <i>op</i> symbol of '+' shall set the appropriate mode bits in the template; '-' shall clear the appropriate bits; '=' shall set the appropriate mode bits, without regard to the contents of process' file mode creation mask. The <i>op</i> symbol of '-' cannot be the first character of <i>mode</i> ; this avoids ambiguity with the optional leading hyphen. Since the initial mode is all bits off, there are not any symbolic modes that need to use '-' as the first character. |
| 18521 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18522 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18523 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18524 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18525 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18526 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18527 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18528 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18529 |                      | If the hyphen is omitted, the primary shall evaluate as true when the file permission bits exactly match the value of the resulting template.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 18530 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18531 |                      | Otherwise, if <i>mode</i> is prefixed by a hyphen, the primary shall evaluate as true if at least all the bits in the resulting template are set in the file permission bits.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 18532 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18533 | <b>-perm [-]onum</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18534 | MAN                  | If the hyphen is omitted, the primary shall evaluate as true when the file permission bits exactly match the value of the octal number <i>onum</i> and only the bits corresponding to the octal mask 07777 shall be compared. (See the description of the octal <i>mode</i> in <i>chmod</i> on page 273.) Otherwise, if <i>onum</i> is prefixed by a hyphen, the primary shall evaluate as true if at least all of the bits specified in <i>onum</i> that are also set in the octal mask 07777 are set.                                                                                                                                                                                                                                                                          |
| 18535 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18536 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18537 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18538 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18539 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18540 | <b>-type c</b>       | The primary shall evaluate as true if the type of the file is <i>c</i> , where <i>c</i> is 'b', 'c', 'd', 'l', 'p', or 'f' for block special file, character special file, directory, symbolic link, FIFO, or regular file, respectively.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 18541 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18542 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18543 | <b>-links n</b>      | The primary shall evaluate as true if the file has <i>n</i> links.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 18544 | <b>-user uname</b>   | The primary shall evaluate as true if the file belongs to the user <i>uname</i> . If <i>uname</i> is a decimal integer and the <i>getpwnam()</i> (or equivalent) function does not return a valid user name, <i>uname</i> shall be interpreted as a user ID.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 18545 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18546 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18547 | <b>-group gname</b>  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18548 |                      | The primary shall evaluate as true if the file belongs to the group <i>gname</i> . If <i>gname</i> is a decimal integer and the <i>getgrnam()</i> (or equivalent) function does not return a valid group name, <i>gname</i> shall be interpreted as a group ID.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18549 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 18550 |                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

18551       **-size** *n*[*c*]     The primary shall evaluate as true if the file size in bytes, divided by 512 and  
18552                               rounded up to the next integer, is *n*. If *n* is followed by the character '*c*', the size  
18553                               shall be in bytes.

18554       **-atime** *n*       The primary shall evaluate as true if the file access time subtracted from the  
18555                               initialization time, divided by 86 400 (with any remainder discarded), is *n*.

18556       **-ctime** *n*       The primary shall evaluate as true if the time of last change of file status  
18557                               information subtracted from the initialization time, divided by 86 400 (with any  
18558                               remainder discarded), is *n*.

18559       **-mtime** *n*       The primary shall evaluate as true if the file modification time subtracted from the  
18560                               initialization time, divided by 86 400 (with any remainder discarded), is *n*.

18561       **-exec** *utility\_name* [*argument* . . .];  
18562                               The primary shall evaluate as true if the invoked utility *utility\_name* returns a zero  
18563                               value as exit status. The end of the primary expression shall be punctuated by a  
18564                               semicolon. A *utility\_name* or *argument* containing only the two characters "{}"  
18565                               shall be replaced by the current path name. If a *utility\_name* or argument string  
18566                               contains the two characters "{}", but not just the two characters "{}", it is  
18567                               implementation-dependent whether *find* replaces those two characters with the  
18568                               current path name or uses the string without change. The current directory for the  
18569                               invocation of *utility\_name* shall be the same as the current directory when the *find*  
18570                               utility was started. If the *utility\_name* names any of the special built-in utilities in  
18571                               Section 2.14 on page 96, the results are undefined.

18572       **-ok** *utility\_name* [*argument* . . .];  
18573                               The **-ok** primary shall be equivalent to **-exec**, except that *find* shall request  
18574                               affirmation of the invocation of *utility\_name* using the current file as an argument  
18575                               by writing to standard error as described in the STDERR section. If the response on  
18576                               standard input is affirmative, the utility shall be invoked. Otherwise, the command  
18577                               shall not be invoked and the value of the **-ok** operand shall be false.

18578       **-print**           The primary always shall evaluate as true; it shall cause the current path name to  
18579                               be written to standard output.

18580       **-newer** *file*     The primary shall evaluate as true if the modification time of the current file is  
18581                               more recent than the modification time of the file named by the path name *file*.

18582       **-depth**           The primary shall always evaluate as true; it shall cause descent of the directory  
18583                               hierarchy to be done so that all entries in a directory are acted on before the  
18584                               directory itself. If a **-depth** primary is not specified, all entries in a directory shall  
18585                               be acted on after the directory itself. If any **-depth** primary is specified, it shall  
18586                               apply to the entire expression even if the **-depth** primary would not normally be  
18587                               evaluated.

18588       The primaries can be combined using the following operators (in order of decreasing  
18589       precedence):

18590       (*expression*)    True if *expression* is true.

18591       !*expression*   Negation of a primary; the unary NOT operator.

18592       *expression* [**-a**] *expression*  
18593                               Conjunction of primaries; the AND operator is implied by the juxtaposition of two  
18594                               primaries or made explicit by the optional **-a** operator. The second expression  
18595                               shall not be evaluated if the first expression is false.

|       |                                               |                                                                                                                            |
|-------|-----------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| 18596 | <i>expression</i> <b>-o</b> <i>expression</i> |                                                                                                                            |
| 18597 |                                               | Alternation of primaries; the OR operator. The second expression shall not be                                              |
| 18598 |                                               | evaluated if the first expression is true.                                                                                 |
| 18599 |                                               | If no <i>expression</i> is present, <b>-print</b> shall be used as the expression. Otherwise, if the given                 |
| 18600 |                                               | expression does not contain any of the primaries <b>-exec</b> , <b>-ok</b> , or <b>-print</b> , the given expression shall |
| 18601 |                                               | be effectively replaced by:                                                                                                |
| 18602 | ( <i>given_expression</i> ) <b>-print</b>     |                                                                                                                            |
| 18603 |                                               | The <b>-user</b> , <b>-group</b> , and <b>-newer</b> primaries each shall evaluate their respective arguments only         |
| 18604 |                                               | once.                                                                                                                      |
| 18605 | <b>STDIN</b>                                  |                                                                                                                            |
| 18606 |                                               | If the <b>-ok</b> primary is used, the response shall be read from the standard input. An entire line                      |
| 18607 |                                               | shall be read as the response. Otherwise, the standard input shall not be used.                                            |
| 18608 | <b>INPUT FILES</b>                            |                                                                                                                            |
| 18609 |                                               | None.                                                                                                                      |
| 18610 | <b>ENVIRONMENT VARIABLES</b>                  |                                                                                                                            |
| 18611 |                                               | The following environment variables shall affect the execution of <i>find</i> :                                            |
| 18612 | <i>LANG</i>                                   | Provide a default value for the internationalization variables that are unset or null.                                     |
| 18613 |                                               | If <i>LANG</i> is unset or null, the corresponding value from the implementation-                                          |
| 18614 |                                               | dependent default locale shall be used. If any of the internationalization variables                                       |
| 18615 |                                               | contains an invalid setting, the utility shall behave as if none of the variables had                                      |
| 18616 |                                               | been defined.                                                                                                              |
| 18617 | <i>LC_ALL</i>                                 | If set to a non-empty string value, override the values of all the other                                                   |
| 18618 |                                               | internationalization variables.                                                                                            |
| 18619 | <i>LC_COLLATE</i>                             |                                                                                                                            |
| 18620 |                                               | Determine the locale for the behavior of ranges, equivalence classes and multi-                                            |
| 18621 |                                               | character collating elements used in the pattern matching notation for the <b>-n</b>                                       |
| 18622 |                                               | option and in the extended regular expression defined for the <b>yesexpr</b> locale                                        |
| 18623 |                                               | keyword in the <i>LC_MESSAGES</i> category.                                                                                |
| 18624 | <i>LC_CTYPE</i>                               | This variable determines the locale for the interpretation of sequences of bytes of                                        |
| 18625 |                                               | text data as characters (for example, single-byte as opposed to multi-byte                                                 |
| 18626 |                                               | characters in arguments), the behavior of character classes within the pattern                                             |
| 18627 |                                               | matching notation used for the <b>-n</b> option, and the behavior of character classes                                     |
| 18628 |                                               | within regular expressions used in the extended regular expression defined for the                                         |
| 18629 |                                               | <b>yesexpr</b> locale keyword in the <i>LC_MESSAGES</i> category.                                                          |
| 18630 | <i>LC_MESSAGES</i>                            |                                                                                                                            |
| 18631 |                                               | Determine the locale for the processing of affirmative responses that should be                                            |
| 18632 |                                               | used to affect the format and contents of diagnostic messages written to standard                                          |
| 18633 |                                               | error.                                                                                                                     |
| 18634 | XSI <i>NLSPATH</i>                            | Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .                                      |
| 18635 | <i>PATH</i>                                   | Determine the location of the <i>utility_name</i> for the <b>-exec</b> and <b>-ok</b> primaries, as                        |
| 18636 |                                               | described in the System Interface Definitions volume of IEEE Std. 1003.1-200x,                                             |
| 18637 |                                               | Chapter 8, Environment Variables.                                                                                          |

18638 **ASYNCHRONOUS EVENTS**

18639 Default.

18640 **STDOUT**

18641 The **-print** primary shall cause the current path names to be written to standard output. The  
 18642 format shall be:

18643 "%s\n", &lt;path&gt;

18644 **STDERR**

18645 The **-ok** primary shall write a prompt to standard error containing at least the *utility\_name* to be  
 18646 invoked and the current path name. In the POSIX locale, the last non-<blank> character in the  
 18647 prompt shall be '?'. The exact format used is unspecified.

18648 Otherwise, the standard error shall be used only for diagnostic messages.

18649 **OUTPUT FILES**

18650 None.

18651 **EXTENDED DESCRIPTION**

18652 None.

18653 **EXIT STATUS**

18654 The following exit values shall be returned:

18655 0 All *path* operands were traversed successfully.

18656 &gt;0 An error occurred.

18657 **CONSEQUENCES OF ERRORS**

18658 Default.

18659 **APPLICATION USAGE**

18660 When used in operands, pattern matching notation, semicolons, opening parentheses, and  
 18661 closing parentheses are special to the shell and must be quoted (see Section 2.2 on page 36).

18662 The bit that is traditionally used for sticky (historically 01000) is specified in the **-perm** primary  
 18663 using the octal number argument form. Since this bit is not defined by this volume of  
 18664 IEEE Std. 1003.1-200x, applications must not assume that it actually refers to the traditional  
 18665 sticky bit.

18666 **EXAMPLES**

18667 1. The following commands are equivalent:

18668 find .

18669 find . -print

18670 They both write out the entire directory hierarchy from the current directory.

18671 2. The following command:

18672 find / \( -name tmp -o -name '\*.xx' \) -atime +7 -exec rm {} \;

18673 removes all files named **tmp** or ending in **.xx** that have not been accessed for seven or more  
 18674 24-hour periods.

18675 3. The following command:

18676 find . -perm -o+w,+s

18677 prints (**-print** is assumed) the names of all files in or below the current directory, with all  
 18678 of the file permission bits **S\_ISUID**, **S\_ISGID**, and **S\_IWOTH** set.

4. The following command:

```
find . -name SCCS -prune -o -print
```

recursively prints path names of all files in the current directory and below, but skips directories named SCCS and files in them.

5. The following command:

```
find . -print -name SCCS -prune
```

behaves as in the previous example, but prints the names of the SCCS directories.

6. The following command is roughly equivalent to the **-nt** extension to *test*:

```
if [-n "$(find file1 -prune -newer file2)"]; then
 printf %s\\n "file1 is newer than file2"
fi
```

7. The descriptions of **-atime**, **-ctime**, and **-mtime** use the terminology *n* “24-hour periods”. For example, a file accessed at 23:59 is selected by:

```
find . -atime -1 -print
```

at 00:01 the next day (less than 24 hours later, not more than one day ago); the midnight boundary between days has no effect on the 24-hour calculation.

#### RATIONALE

The **-a** operator was retained as an optional operator for compatibility with historical shell scripts, even though it is redundant with expression concatenation.

The descriptions of the ‘-’ modifier on the *mode* and *onum* arguments to the **-perm** primary agree with historical practice on BSD and System V implementations. System V and BSD documentation both describe it in terms of checking additional bits; in fact, it uses the same bits, but checks for having at least all of the matching bits set instead of having exactly the matching bits set.

The exact format of the interactive prompts is unspecified. Only the general nature of the contents of prompts are specified because:

- Implementations may desire more descriptive prompts than those used on historical implementations.
- Since the historical prompt strings do not terminate with <newline>s, there is no portable way for another program to interact with the prompts of this utility via pipes.

Therefore, an application using this prompting option relies on the system to provide the most suitable dialog directly with the user, based on the general guidelines specified.

The **-name file** operand was changed to use the shell pattern matching notation so that *find* is consistent with other utilities using pattern matching.

For the **-type c** operand, implementors of symbolic links should consider **l** (the letter ell) for symbolic links. Implementations that support sockets also use **-type s** for sockets. Implementations planning to add options to allow *find* to follow symbolic links or treat them as special files should consider the **-follow** primary implemented in BSD and System V Release 4 as a guide.

The **-size** operand refers to the size of a file, rather than the number of blocks it may occupy in the file system. The intent is that the *st\_size* field defined in the System Interfaces volume of IEEE Std. 1003.1-200x should be used, not the *st\_blocks* found in historical implementations. There are at least two reasons for this:

1. In both System V and BSD, *find* only uses *st\_size* in size calculations for the operands specified by this volume of IEEE Std. 1003.1-200x. (BSD uses *st\_blocks* only when processing the *-ls* primary.)
2. Users usually think of file size in terms of bytes, which is also the unit used by the *ls* utility for the output from the *-l* option. (In both System V and BSD, *ls* uses *st\_size* for the *-l* option size field and uses *st\_blocks* for the *ls -s* calculations. This volume of IEEE Std. 1003.1-200x does not specify *ls -s*.)

The descriptions of *-atime*, *-ctime*, and *-mtime* were changed from the SVID description of *n* “days” to “24-hour periods”. The description is also different in terms of the exact timeframe for the *n* case (*versus* the *+n* or *-n*), but it matches all known historical implementations. It refers to one 24-hour period in the past, not any time from the beginning of that period to the current time. For example, *-atime 3* is true if the file was accessed any time in the period from 72 hours to 48 hours ago.

Historical implementations do not modify “{ }” when it appears as a substring of an *-exec* or *-ok utility\_name* or argument string. There have been numerous user requests for this extension, so this volume of IEEE Std. 1003.1-200x allows the desired behavior. At least one recent implementation does support this feature, but encountered several problems in managing memory allocation and dealing with multiple occurrences of “{ }” in a string while it was being developed, so it is not yet required behavior.

Assuming the presence of *-print* was added to correct a historical pitfall that plagues novice users, it is entirely upward-compatible from the historical System V *find* utility. In its simplest form (*find directory*), it could be confused with the historical BSD fast *find*. The BSD developers agreed that adding *-print* as a default expression was the correct decision and have added the fast *find* functionality within a new utility called *locate*.

Historically, the *-L* option was implemented using the primary *-follow*. The *-H* and *-L* options were added for two reasons. First, they offer a finer granularity of control and consistency with other programs that walk file hierarchies. Second, the *-follow* primary always evaluated to true. As they were historically really global variables that took effect before the traversal began, some valid expressions had unexpected results. An example is the expression *-print -o -follow*. Because *-print* always evaluates to true, the standard order of evaluation implies that *-follow* would never be evaluated. This was never the case. Historical practice for the *-follow* primary, however, is not consistent. Some implementations always follow symbolic links on the command line whether *-follow* is specified or not. Others follow symbolic links on the command line only if *-follow* is specified. Both behaviors are provided by the *-H* and *-L* options, but scripts using the current *-follow* primary would be broken if the *-follow* option is specified to work either way.

Since the *-L* option resolves all symbolic links and the *-type l* primary is true for symbolic links that still exist after symbolic links have been resolved, the command:

```
find -L . -type l
```

prints a list of symbolic links reachable from the current directory that do not resolve to accessible files.

## FUTURE DIRECTIONS

None.

## SEE ALSO

*chmod*, *pax*, *sh*, *test*, the System Interfaces volume of IEEE Std. 1003.1-200x, *stat()*



18767 **CHANGE HISTORY**

18768 First released in Issue 2.

18769 **Issue 4**

18770 Aligned with the ISO/IEC 9945-2: 1993 standard.

18771 **Issue 5**

18772 FUTURE DIRECTIONS section added.

18773 **Issue 6**

18774 The following new requirements on POSIX implementations derive from alignment with the  
18775 Single UNIX Specification:

18776 • The **-perm [-]onum** primary is supported.

18777 The *find* utility is aligned with the IEEE P1003.2b draft standard, to include processing of  
18778 symbolic links and changes to the description of the **atime**, **ctime**, and **mtime** operands.

## 18779 NAME

18780 fold — filter for folding lines

## 18781 SYNOPSIS

18782 fold [-bs][-w *width*][*file...*]

## 18783 DESCRIPTION

18784 The *fold* utility is a filter that shall fold lines from its input files, breaking the lines to have a  
 18785 maximum of *width* column positions (or bytes, if the **-b** option is specified). Lines shall be  
 18786 broken by the insertion of a <newline> character such that each output line (referred to later in  
 18787 this section as a *segment*) is the maximum width possible that does not exceed the specified  
 18788 number of column positions (or bytes). A line shall not be broken in the middle of a character.  
 18789 The behavior is undefined if *width* is less than the number of columns any single character in the  
 18790 input would occupy.

18791 If the <carriage-return>, <backspace>, or <tab> characters are encountered in the input, and the  
 18792 **-b** option is not specified, they shall be treated specially:

18793 <backspace> The current count of line width shall be decremented by one, although the count  
 18794 never shall become negative. The *fold* utility shall not insert a <newline> character  
 18795 immediately before or after any <backspace> character.

18796 <carriage-return>  
 18797 The current count of line width shall be set to zero. The *fold* utility shall not insert a  
 18798 <newline> character immediately before or after any <carriage-return> character.

18799 <tab> Each <tab> character encountered shall advance the column position pointer to the  
 18800 next tab stop. Tab stops shall be at each column position *n* such that *n* modulo 8  
 18801 equals 1.

## 18802 OPTIONS

18803 The *fold* utility shall conform to the System Interface Definitions volume of  
 18804 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

18805 The following options shall be supported:

18806 **-b** Count *width* in bytes rather than column positions.

18807 **-s** If a segment of a line contains a <blank> character within the first *width* column  
 18808 positions (or bytes), break the line after the last such <blank> character meeting the  
 18809 width constraints. If there is no <blank> character meeting the requirements, the **-s**  
 18810 option shall have no effect for that output segment of the input line.

18811 **-w *width*** Specify the maximum line length, in column positions (or bytes if **-b** is specified).  
 18812 The results are unspecified if *width* is not a positive decimal number. The default  
 18813 value shall be 80.

## 18814 OPERANDS

18815 The following operand shall be supported:

18816 *file* A path name of a text file to be folded. If no *file* operands are specified, the  
 18817 standard input shall be used.

## 18818 STDIN

18819 The standard input shall be used only if no *file* operands are specified. See the INPUT FILES  
 18820 section.

18821 **INPUT FILES**

18822       If the **-b** option is specified, the input files shall be text files except that the lines are not limited  
 18823       to {LINE\_MAX} bytes in length. If the **-b** option is not specified, the input files shall be text files.

18824 **ENVIRONMENT VARIABLES**

18825       The following environment variables shall affect the execution of *fold*:

18826       **LANG**       Provide a default value for the internationalization variables that are unset or null.  
 18827                      If *LANG* is unset or null, the corresponding value from the implementation-  
 18828                      dependent default locale shall be used. If any of the internationalization variables  
 18829                      contains an invalid setting, the utility shall behave as if none of the variables had  
 18830                      been defined.

18831       **LC\_ALL**       If set to a non-empty string value, override the values of all the other  
 18832                      internationalization variables.

18833       **LC\_CTYPE**   Determine the locale for the interpretation of sequences of bytes of text data as  
 18834                      characters (for example, single-byte as opposed to multi-byte characters in  
 18835                      arguments and input files), and for the determination of the width in column  
 18836                      positions each character would occupy on a constant-width font output device.

18837       **LC\_MESSAGES**  
 18838                      Determine the locale that should be used to affect the format and contents of  
 18839                      diagnostic messages written to standard error.

18840 **XSI**       **NLSPATH**   Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

18841 **ASYNCHRONOUS EVENTS**

18842       Default.

18843 **STDOUT**

18844       The standard output shall be a file containing a sequence of characters whose order shall be  
 18845       preserved from the input files, possibly with inserted <newline> characters.

18846 **STDERR**

18847       Used only for diagnostic messages.

18848 **OUTPUT FILES**

18849       None.

18850 **EXTENDED DESCRIPTION**

18851       None.

18852 **EXIT STATUS**

18853       The following exit values shall be returned:

18854       0   All input files were processed successfully.

18855       >0   An error occurred.

18856 **CONSEQUENCES OF ERRORS**

18857       Default.

**18858 APPLICATION USAGE**

18859       The *cut* and *fold* utilities can be used to create text files out of files with arbitrary line lengths. The  
18860       *cut* utility should be used when the number of lines (or records) needs to remain constant. The  
18861       *fold* utility should be used when the contents of long lines need to be kept contiguous.

18862       The *fold* utility is frequently used to send text files to printers that truncate, rather than fold, lines  
18863       wider than the printer is able to print (usually 80 or 132 column positions).

**18864 EXAMPLES**

18865       An example invocation that submits a file of possibly long lines to the printer (under the  
18866       assumption that the user knows the line width of the printer to be assigned by *lp*):

18867       fold -w 132 bigfile | lp

**18868 RATIONALE**

18869       Although terminal input in canonical processing mode requires the erase character (frequently  
18870       set to <backspace>) to erase the previous character (not byte or column position), terminal  
18871       output is not buffered and is extremely difficult, if not impossible, to parse correctly; the  
18872       interpretation depends entirely on the physical device that actually displays/prints/stores the  
18873       output. In all known internationalized implementations, the utilities producing output for mixed  
18874       column-width output assume that a <backspace> backs up one column position and outputs  
18875       enough <backspace>s to return to the start of the character when <backspace> is used to  
18876       provide local line motions to support underlining and emboldening operations. Since *fold*  
18877       without the *-b* option is dealing with these same constraints, <backspace> is always treated as  
18878       backing up one column position rather than backing up one character.

18879       Historical versions of the *fold* utility assumed 1 byte was one character and occupied one column  
18880       position when written out. This is no longer always true. Since the most common usage of *fold* is  
18881       believed to be folding long lines for output to limited-length output devices, this capability was  
18882       preserved as the default case. The *-b* option was added so that applications could *fold* files with  
18883       arbitrary length lines into text files that could then be processed by the standard utilities. Note  
18884       that although the width for the *-b* option is in bytes, a line is never split in the middle of a  
18885       character. (It is unspecified what happens if a width is specified that is too small to hold a single  
18886       character found in the input followed by a <newline>.)

18887       The tab stops are hardcoded to be every eighth column to meet historical practice. No new  
18888       method of specifying other tab stops was invented.

**18889 FUTURE DIRECTIONS**

18890       None.

**18891 SEE ALSO**

18892       *cut*

**18893 CHANGE HISTORY**

18894       First released in Issue 4.

**18895 Issue 6**

18896       The normative text is reworded to avoid use of the term “must” for application requirements.

18897 **NAME**18898 fort77 — FORTRAN compiler (**FORTTRAN**)18899 **SYNOPSIS**

```
18900 FD fort77 [-c][-g][-L directory]... [-O optlevel][-o outfile][-s][-w]
18901 operand...
```

18902

18903 **DESCRIPTION**

18904 The *fort77* utility is the interface to the FORTRAN compilation system; it shall accept the full  
 18905 FORTRAN-77 language defined by the ANSI X3.9-1978 standard. The system conceptually  
 18906 consists of a compiler and link editor. The files referenced by *operands* are compiled and linked  
 18907 to produce an executable file. It is unspecified whether the linking occurs entirely within the  
 18908 operation of *fort77*; some systems may produce objects that are not fully resolved until the file is  
 18909 executed.

18910 If the **-c** option is present, for all path name operands of the form *file.f*, the files:

18911 \$(basename *pathname.f*).o

18912 shall be created or overwritten as the result of successful compilation. If the **-c** option is not  
 18913 specified, it is unspecified whether such .o files are created or deleted for the *file.f* operands.

18914 If there are no options that prevent link editing (such as **-c**) and all operands compile and link  
 18915 without error, the resulting executable file shall be written into the file named by the **-o** option  
 18916 (if present) or to the file **a.out**. The executable file shall be created as specified in the System  
 18917 Interfaces volume of IEEE Std. 1003.1-200x, except that the file permissions shall be set to:

18918 S\_IRWXO | S\_IRWXG | S\_IRWXU

18919 and that the bits specified by the *umask* of the process shall be cleared.

18920 **OPTIONS**

18921 The *fort77* utility shall conform to the System Interface Definitions volume of  
 18922 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines, except that:

- 18923 • The **-l** *library* operands have the format of options, but their position within a list of  
 18924 operands affects the order in which libraries are searched.
- 18925 • The order of specifying the multiple **-L** options is significant.
- 18926 • Portable applications shall specify each option separately; that is, grouping option letters (for  
 18927 example, **-cg**) need not be recognized by all implementations.

18928 The following options shall be supported:

18929 **-c** Suppress the link-edit phase of the compilation, and do not remove any object files  
 18930 that are produced.

18931 **-g** Produce symbolic information in the object or executable files; the nature of this  
 18932 information is unspecified, and may be modified by implementation-dependent  
 18933 interactions with other options.

18934 **-s** Produce object or executable files, or both, from which symbolic and other  
 18935 information not required for proper execution using the *exec* family of functions  
 18936 defined in the System Interfaces volume of IEEE Std. 1003.1-200x has been  
 18937 removed (stripped). If both **-g** and **-s** options are present, the action taken is  
 18938 unspecified.

18939 **-o** *outfile* Use the path name *outfile*, instead of the default **a.out**, for the executable file  
 18940 produced. If the **-o** option is present with **-c**, the result is unspecified.

18941        **-L *directory***    Change the algorithm of searching for the libraries named in **-l** operands to look in  
 18942                            the directory named by the *directory* path name before looking in the usual places.  
 18943                            Directories named in **-L** options shall be searched in the specified order. At least  
 18944                            ten instances of this option shall be supported in a single *fort77* command  
 18945                            invocation. If a directory specified by a **-L** option contains a file named **libf.a**, the  
 18946                            results are unspecified.

18947        **-O *optlevel***    Specify the level of code optimization. If the *optlevel* option-argument is the digit  
 18948                            '0', all special code optimizations shall be disabled. If it is the digit '1', the  
 18949                            nature of the optimization is unspecified. If the **-O** option is omitted, the nature of  
 18950                            the system's default optimization is unspecified. It is unspecified whether code  
 18951                            generated in the presence of the **-O 0** option is the same as that generated when  
 18952                            **-O** is omitted. Other *optlevel* values may be supported.

18953        **-w**                Suppress warnings.

18954        Multiple instances of **-L** options can be specified.

## 18955 OPERANDS

18956        An *operand* is either in the form of a path name or the form **-l *library***. At least one operand of the  
 18957        path name form shall be specified. The following operands shall be supported:

18958        ***file.f***            The path name of a FORTRAN source file to be compiled and optionally passed to  
 18959                            the link editor. The file name operand shall be of this form if the **-c** option is used.

18960        ***file.a***            A library of object files typically produced by *ar*, and passed directly to the link  
 18961                            editor. Implementations may recognize implementation-dependent suffixes other  
 18962                            than **.a** as denoting object file libraries.

18963        ***file.o***            An object file produced by *fort77 -c* and passed directly to the link editor.  
 18964                            Implementations may recognize implementation-dependent suffixes other than **.o**  
 18965                            as denoting object files.

18966        The processing of other files is implementation-dependent.

18967        **-l *library***        (The letter ell.) Search the library named:

18968                            *liblibrary.a*

18969                            A library is searched when its name is encountered, so the placement of a **-l**  
 18970                            operand is significant. Several standard libraries can be specified in this manner, as  
 18971                            described in the EXTENDED DESCRIPTION section. Implementations may  
 18972                            recognize implementation-dependent suffixes other than **.a** as denoting libraries.

## 18973 STDIN

18974        Not used.

## 18975 INPUT FILES

18976        The input file shall be one of the following: a text file containing FORTRAN source code; an  
 18977        object file in the format produced by *fort77 -c*; or a library of object files, in the format produced  
 18978        by archiving zero or more object files, using *ar*. Implementations may supply additional utilities  
 18979        that produce files in these formats. Additional input files are implementation-dependent.

18980        A <tab> character encountered within the first six characters on a line of source code shall cause  
 18981        the compiler to interpret the following character as if it were the seventh character on the line  
 18982        (that is, in column 7).

18983 **ENVIRONMENT VARIABLES**

18984 The following environment variables shall affect the execution of *fort77*:

18985 **LANG** Provide a default value for the internationalization variables that are unset or null.  
 18986 If *LANG* is unset or null, the corresponding value from the implementation-  
 18987 dependent default locale shall be used. If any of the internationalization variables  
 18988 contains an invalid setting, the utility shall behave as if none of the variables had  
 18989 been defined.

18990 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 18991 internationalization variables.

18992 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 18993 characters (for example, single-byte as opposed to multi-byte characters in  
 18994 arguments and input files).

18995 **LC\_MESSAGES**  
 18996 Determine the locale that should be used to affect the format and contents of  
 18997 diagnostic messages written to standard error.

18998 **XS1** **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

18999 **TMPDIR** Determine the path name that should override the default directory for temporary  
 19000 files, if any.

19001 **ASYNCHRONOUS EVENTS**

19002 Default.

19003 **STDOUT**

19004 Not used.

19005 **STDERR**

19006 Used only for diagnostic messages. If more than one file operand ending in *.f* (or possibly other  
 19007 unspecified suffixes) is given, for each such file:

19008 "%s:\n", <file>

19009 may be written to allow identification of the diagnostic message with the appropriate input file.

19010 This utility may produce warning messages about certain conditions that do not warrant  
 19011 returning an error (non-zero) exit value.

19012 **OUTPUT FILES**

19013 Object files, listing files and executable files shall be produced in unspecified formats.

19014 **EXTENDED DESCRIPTION**19015 **Standard Libraries**

19016 The *fort77* utility shall recognize the following **-l** operand for the standard library:

19017 **-l f** This library contains all library functions referenced in the ANSI X3.9-1978  
 19018 standard. This operand shall not be required to be present to cause a search of this  
 19019 library.

19020 In the absence of options that inhibit invocation of the link editor, such as **-c**, the *fort77* utility  
 19021 shall cause the equivalent of a **-l f** operand to be passed to the link editor as the last **-l** operand,  
 19022 causing it to be searched after all other object files and libraries are loaded.

19023 It is unspecified whether the library **libf.a** exists as a regular file. The implementation may  
 19024 accept as **-l** operands names of objects that do not exist as regular files.

19025 **External Symbols**

19026 The FORTRAN compiler and link editor shall support the significance of external symbols up to  
 19027 a length of at least 31 bytes; case folding is permitted. The action taken upon encountering  
 19028 symbols exceeding the implementation-dependent maximum symbol length is unspecified.

19029 The compiler and link editor shall support a minimum of 511 external symbols per source or  
 19030 object file, and a minimum of 4 095 external symbols total. A diagnostic message is written to  
 19031 standard output if the implementation-dependent limit is exceeded; other actions are  
 19032 unspecified.

19033 **EXIT STATUS**

19034 The following exit values shall be returned:

19035     0 Successful compilation or link edit.

19036     >0 An error occurred.

19037 **CONSEQUENCES OF ERRORS**

19038 When *fort77* encounters a compilation error, it shall write a diagnostic to standard error and  
 19039 continue to compile other source code operands. It shall return a non-zero exit status, but it is  
 19040 implementation-dependent whether an object module is created. If the link edit is unsuccessful,  
 19041 a diagnostic message shall be written to standard error, and *fort77* shall exit with a non-zero  
 19042 status.

19043 **APPLICATION USAGE**

19044 None.

19045 **EXAMPLES**

19046 The following usage example compiles **xyz.f** and creates the executable file **foo**:

19047 `fort77 -o foo xyz.f`

19048 The following example compiles **xyz.f** and creates the object file **xyz.o**:

19049 `fort77 -c xyz.f`

19050 The following example compiles **xyz.f** and creates the executable file **a.out**:

19051 `fort77 xyz.f`

19052 The following example compiles **xyz.f**, links it with **b.o**, and creates the executable **a.out**:

19053 `fort77 xyz.f b.o`

19054 **RATIONALE**

19055 The name of this utility was chosen as *fort77* to parallel the renaming of the C compiler. The  
 19056 name *f77* was not chosen to avoid problems with historical implementations. The  
 19057 ANSI X3.9-1978 standard was selected as a normative reference because the ISO/IEC version of  
 19058 FORTRAN-77 has been superseded by the ISO/IEC 1539: 1990 standard (Fortran-90).

19059 The file inclusion and symbol definition **#define** mechanisms used by the *c89* utility were not  
 19060 included in this volume of IEEE Std. 1003.1-200x—even though they are commonly  
 19061 implemented—since there is no requirement that the FORTRAN compiler use the C  
 19062 preprocessor.

19063 The **-onetrip** option was not included in this volume of IEEE Std. 1003.1-200x, even though  
 19064 many historical compilers support it, because it is derived from FORTRAN-66; it is an  
 19065 anachronism that should not be perpetuated.

19066 Some implementations produce compilation listings. This aspect of FORTRAN has been left  
 19067 unspecified because there was controversy concerning the various methods proposed for



19068 implementing it: a `-V` option overlapped with historical vendor practice and a naming  
19069 convention of creating files with `.l` suffixes collided with historical *lex* file naming practice.

19070 There is no `-I` option in this version of this volume of IEEE Std. 1003.1-200x to specify a directory  
19071 for file inclusion. An `INCLUDE` directive has been a part of the Fortran-90 discussions, but an  
19072 interface supporting that standard is not in the current scope.

19073 It is noted that many FORTRAN compilers produce an object module even when compilation  
19074 errors occur; during a subsequent compilation, the compiler may patch the object module rather  
19075 than recompiling all the code. Consequently, it is left to the implementor whether or not an  
19076 object file is created.

19077 A reference to MIL-STD-1753 was removed from an early proposal in response to a request from  
19078 the POSIX FORTRAN-binding standard developers. It was not the intention of the standard  
19079 developers to require certification of the FORTRAN compiler, and the POSIX.9 standard does  
19080 not specify the military standard or any special preprocessing requirements. Furthermore, use of  
19081 that document would have been inappropriate for an international standard.

19082 The specification of optimization has been subject to changes through early proposals. At one  
19083 time, `-O` and `-N` were Booleans: optimize and do not optimize (with an unspecified default).  
19084 Some historical practice lead this to be changed to:

19085 `-O 0`           No optimization.

19086 `-O 1`           Some level of optimization.

19087 `-O n`           Other, unspecified levels of optimization.

19088 It is not always clear whether “good code generation” is the same thing as optimization. Simple  
19089 optimizations of local actions do not usually affect the semantics of a program. The `-O 0` option  
19090 has been included to accommodate the very particular nature of scientific calculations in a  
19091 highly optimized environment; compilers make errors. Some degree of optimization is expected,  
19092 even if it is not documented here, and the ability to shut it off completely could be important  
19093 when porting an application. An implementation may treat `-O 0` as “do less than normal” if it  
19094 wishes, but this is only meaningful if any of the operations it performs can affect the semantics  
19095 of a program. It is highly dependent on the implementation whether doing less than normal is  
19096 logical. It is not the intent of the `-O 0` option to ask for inefficient code generation, but rather to  
19097 assure that any semantically visible optimization is suppressed.

19098 The specification of standard library access is consistent with the C compiler specification.  
19099 Implementations are not required to have `/usr/lib/libf.a`, as many historical implementations do,  
19100 but if not they are required to recognize `f` as a token.

19101 External symbol size limits are in normative text; portable applications need to know these  
19102 limits. However, the minimum maximum symbol length should be taken as a constraint on a  
19103 portable application, not on an implementation, and consequently the action taken for a symbol  
19104 exceeding the limit is unspecified. The minimum size for the external symbol table was added  
19105 for similar reasons.

19106 The CONSEQUENCES OF ERRORS section clearly specifies the behavior of the compiler when  
19107 compilation or link-edit errors occur. The behavior of several historical implementations was  
19108 examined, and the choice was made to be silent on the status of the executable, or `a.out`, file in  
19109 the face of compiler or linker errors. If a linker writes the executable file, then links it on disk  
19110 with `lseek()`s and `write()`s, the partially linked executable file can be left on disk and its execute  
19111 bits turned off if the link edit fails. However, if the linker links the image in memory before  
19112 writing the file to disk, it need not touch the executable file (if it already exists) because the link  
19113 edit fails. Since both approaches are historical practice, a portable application shall rely on the  
19114 exit status of *fort77*, rather than on the existence or mode of the executable file.

19115 The `-g` and `-s` options are not specified as mutually-exclusive. Historically these two options  
19116 have been mutually-exclusive, but because both are so loosely specified, it seemed appropriate  
19117 to leave their interaction unspecified.

19118 The requirement that portable applications specify compiler options separately is to reserve the  
19119 multi-character option namespace for vendor-specific compiler options, which are known to  
19120 exist in many historical implementations. Implementations are not required to recognize, for  
19121 example, `-gc` as if it were `-g -c`; nor are they forbidden from doing so. The SYNOPSIS shows all  
19122 of the options separately to highlight this requirement on applications.

19123 Echoing file names to standard error is considered a diagnostic message because it would  
19124 otherwise be difficult to associate an error message with the erring file. They are described with  
19125 “may” to allow implementations to use other methods of identifying files and to parallel the  
19126 description in *c89*.

#### 19127 FUTURE DIRECTIONS

19128 A compilation system based on the ISO/IEC 1539: 1990 standard (Fortran-90) shall be considered  
19129 for a future issue; it may have a different utility name from *fort77*.

#### 19130 SEE ALSO

19131 *ar*, *asa*, *c89*, *umask*

#### 19132 CHANGE HISTORY

19133 First released in Issue 4.

#### 19134 Issue 6

19135 This utility is now marked as part of the FORTRAN Development Utilities option.

19136 The normative text is reworded to avoid use of the term “must” for application requirements.

19137 **NAME**

19138           fuser — list process IDs of all processes that have one or more files open

19139 **SYNOPSIS**

19140 xSI       fuser [ -cfu ] *file* ...

19141

19142 **DESCRIPTION**

19143           The *fuser* utility shall write to standard output the process IDs of processes running on the local  
19144           system that have one or more named files open. For block special devices, all processes using  
19145           any file on that device are listed.

19146           The *fuser* utility shall write to standard error additional information about the named files  
19147           indicating how the file is being used.

19148           Any output for processes running on remote systems that have a named file open is unspecified.

19149           A user may need appropriate privilege to invoke the *fuser* utility.

19150 **OPTIONS**

19151           The *fuser* utility shall conform to the System Interface Definitions volume of  
19152           IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

19153           The following options shall be supported:

19154           -c           The file is treated as a mount point and the utility shall report on any files open in  
19155           the file system.

19156           -f           The report shall be only for the named files.

19157           -u           The user name, in parentheses, associated with each process ID written to standard  
19158           output shall be written to standard error.

19159 **OPERANDS**

19160           The following operand shall be supported:

19161           *file*       A path name on which the file or file system is to be reported.

19162 **STDIN**

19163           Not used.

19164 **INPUT FILES**

19165           The user database.

19166 **ENVIRONMENT VARIABLES**

19167           The following environment variables shall affect the execution of *fuser*:

19168           *LANG*       Provide a default value for the internationalization variables that are unset or null.  
19169                       If *LANG* is unset or null, the corresponding value from the implementation-  
19170                       dependent default locale shall be used. If any of the internationalization variables  
19171                       contain an invalid setting, the utility behaves as if none of the variables had been  
19172                       set.

19173           *LC\_ALL*     If set to a non-empty string value, override the values of all the other  
19174           internationalization variables.

19175           *LC\_CTYPE*   Determine the locale for the interpretation of sequences of bytes of text data as  
19176           characters (for example, single-byte as opposed to multi-byte characters in  
19177           arguments).

19178           *LC\_MESSAGES*

19179                       Determine the locale that should be used to affect the format and contents of

19180 diagnostic messages written to standard error.

19181 *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

19182 **ASYNCHRONOUS EVENTS**

19183 Default.

19184 **STDOUT**

19185 The *fuser* utility shall write the process ID for each process using each file given as an operand to

19186 standard output in the following format:

19187 "%d", <process\_id>

19188 **STDERR**

19189 The *fuser* utility shall write diagnostic messages to standard error.

19190 The *fuser* utility also shall write the following to standard error:

19191 • The path name of each named file is written followed immediately by a colon.

19192 • For each process ID written to standard output, the character 'c' shall be written to

19193 standard error if the process is using the file as its current directory and the character 'r'

19194 shall be written to standard error if the process is using the file as its root directory.

19195 Implementations may write other alphabetic characters to indicate other uses of files.

19196 • When the *-u* option is specified, characters indicating the use of the file shall be followed

19197 immediately by the user name, in parentheses, corresponding to the process' real user ID. If

19198 the user name cannot be resolved from the process' real user ID, the process' real user ID

19199 shall be written instead of the user name.

19200 When standard output and standard error are directed to the same file, the output shall be

19201 interspersed so that the file name appears at the start of each line, followed by the process ID

19202 and characters indicating the use of the file. Then, if the *-u* option is specified, the user name or

19203 user ID for each process using that file shall be written.

19204 A <newline> character shall be written to standard error after the last output described above

19205 for each file operand.

19206 **OUTPUT FILES**

19207 None.

19208 **EXTENDED DESCRIPTION**

19209 None.

19210 **EXIT STATUS**

19211 The following exit values shall be returned:

19212 0 Successful completion.

19213 >0 An error occurred.

19214 **CONSEQUENCES OF ERRORS**

19215 Default.

19216 **APPLICATION USAGE**

19217       None.

19218 **EXAMPLES**

19219       The command:

19220       fuser -fu .

19221       writes to standard output the process IDs of processes that are using the current directory and

19222       writes to standard error an indication of how those processes are using the directory and the

19223       user names associated with the processes that are using the current directory.

19224 **RATIONALE**

19225       None.

19226 **FUTURE DIRECTIONS**

19227       None.

19228 **SEE ALSO**

19229       None.

19230 **CHANGE HISTORY**

19231       First released in Issue 5.

19232 **NAME**

19233 gencat — generate a formatted message catalog

19234 **SYNOPSIS**19235 XSI gencat *catfile* *msgfile*...

19236

19237 **DESCRIPTION**

19238 The *gencat* utility shall merge the message text source files *msgfile* into a formatted message  
 19239 catalog *catfile*. The file *catfile* shall be created if it does not already exist. If *catfile* does exist, its  
 19240 messages shall be included in the new *catfile*. If set and message numbers collide, the new  
 19241 message text defined in *msgfile* shall replace the old message text currently contained in *catfile*.

19242 **OPTIONS**

19243 None.

19244 **OPERANDS**

19245 The following operands shall be supported:

19246 *catfile* A path name of the formatted message catalog. If '-' is specified, standard output  
 19247 shall be used. The format of the message catalog produced is unspecified.

19248 *msgfile* A path name of a message text source file. If '-' is specified for an instance of  
 19249 *msgfile*, standard input shall be used. The format of message text source files is  
 19250 defined in the EXTENDED DESCRIPTION section.

19251 **STDIN**19252 The standard input shall not be used unless a *msgfile* operand is specified as '-'.19253 **INPUT FILES**

19254 The input files shall be text files.

19255 **ENVIRONMENT VARIABLES**19256 The following environment variables shall affect the execution of *gencat*:

19257 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 19258 If *LANG* is unset or null, the corresponding value from the implementation-  
 19259 dependent default locale shall be used. If any of the internationalization variables  
 19260 contains an invalid setting, the utility shall behave as if none of the variables had  
 19261 been defined.

19262 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 19263 internationalization variables.

19264 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 19265 characters (for example, single-byte as opposed to multi-byte characters in  
 19266 arguments and input files).

19267 *LC\_MESSAGES*

19268 Determine the locale that should be used to affect the format and contents of  
 19269 diagnostic messages written to standard error.

19270 *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

19271 **ASYNCHRONOUS EVENTS**

19272 Default.

19273 **STDOUT**

19274       The standard output shall not be used unless the *catfile* operand is specified as ‘-’.

19275 **STDERR**

19276       Used only for diagnostic messages.

19277 **OUTPUT FILES**

19278       None.

19279 **EXTENDED DESCRIPTION**

19280       The application shall ensure that the format of a message text source file is defined as follows.

19281       Note that the fields of a message text source line are separated by a single <blank> character.

19282       Any other <blank> characters are considered as being part of the subsequent field.

19283       **\$set *n* comment**

19284               This line specifies the set identifier of the following messages until the next **\$set** or  
 19285               end-of-file appears. The *n* denotes the set identifier, which is defined as a number  
 19286               in the range [1, {NL\_SETMAX}] (see the <limits.h> header defined in the System  
 19287               Interfaces volume of IEEE Std. 1003.1-200x). The application shall ensure that set  
 19288               identifiers are presented in ascending order within a single source file, but need  
 19289               not be contiguous. Any string following the set identifier shall be treated as a  
 19290               comment. If no **\$set** directive is specified in a message text source file, all messages  
 19291               shall be located in an implementation-dependent default message set NL\_SETD  
 19292               (see the <nl\_types.h> header defined in the System Interfaces volume of  
 19293               IEEE Std. 1003.1-200x).

19294       **\$delset *n* comment**

19295               This line deletes message set *n* from an existing message catalog. The *n* denotes the  
 19296               set number [1, {NL\_SETMAX}]. Any string following the set number shall be  
 19297               treated as a comment.

19298       **\$ comment**   A line beginning with ‘\$’ followed by a <blank> character shall be treated as a  
 19299               comment.

19300       ***m* message-text**

19301               The *m* denotes the message identifier, which is defined as a number in the range [1,  
 19302               {NL\_MSGMAX}] (see the <limits.h> defined in the System Interfaces volume of  
 19303               IEEE Std. 1003.1-200x). The *message-text* shall be stored in the message catalog with  
 19304               the set identifier specified by the last **\$set** directive, and with message identifier *m*.  
 19305               If the *message-text* is empty, and a <blank> character field separator is present, an  
 19306               empty string shall be stored in the message catalog. If a message source line has a  
 19307               message number, but neither a field separator nor *message-text*, the existing  
 19308               message with that number (if any) shall be deleted from the catalog. The  
 19309               application shall ensure that message identifiers are in ascending order within a  
 19310               single set, but need not be contiguous. The application shall ensure that the length  
 19311               of *message-text* is in the range [0, {NL\_TEXTMAX}] (see the <limits.h> header  
 19312               defined in the System Interfaces volume of IEEE Std. 1003.1-200x).

19313       **\$quote *n***   This line specifies an optional quote character *c*, which can be used to surround  
 19314               *message-text* so that trailing spaces or null (empty) messages are visible in a  
 19315               message source line. By default, or if an empty **\$quote** directive is supplied, no  
 19316               quoting of *message-text* shall be recognized.

19317       Empty lines in a message text source file shall be ignored. The effects of lines starting with any  
 19318       character other than those defined above are implementation-dependent.

19319           Text strings can contain the special characters and escape sequences defined in the following  
19320           table:

19321  
19322  
19323  
19324  
19325  
19326  
19327  
19328  
19329  
19330

| Description       | Symbol | Sequence |
|-------------------|--------|----------|
| <newline>         | NL(LF) | \n       |
| Horizontal tab    | HT     | \t       |
| <vertical-tab>    | VT     | \v       |
| <backspace>       | BS     | \b       |
| <carriage-return> | CR     | \r       |
| <form-feed>       | FF     | \f       |
| Backslash         | \      | \\       |
| Bit pattern       | ddd    | \ddd     |

19331           The escape sequence "\ddd" consists of backslash followed by one, two, or three octal digits,  
19332           which shall be taken to specify the value of the desired character. If the character following a  
19333           backslash is not one of those specified, the backslash shall be ignored.

19334           Backslash ('\') followed by a <newline> character is also used to continue a string on the  
19335           following line. Thus, the following two lines describe a single message string:

19336  
19337  
19338  
19339

1 This line continues \  
to the next line  
  
which is equivalent to:  
  
1 This line continues to the next line

19340 **EXIT STATUS**

19341           The following exit values shall be returned:

- 19342  
19343
- 0   Successful completion.  
>0  An error occurred.

19344 **CONSEQUENCES OF ERRORS**

19345           Default.

19346 **APPLICATION USAGE**

19347           Message catalogs produced by *gencat* are binary encoded, meaning that their portability cannot  
19348           be guaranteed between different types of machine. Thus, just as C programs need to be  
19349           recompiled for each type of machine, so message catalogs must be recreated via *gencat*.

19350 **EXAMPLES**

19351           None.

19352 **RATIONALE**

19353           None.

19354 **FUTURE DIRECTIONS**

19355           None.

19356 **SEE ALSO**

19357           *iconv*, the System Interfaces volume of IEEE Std. 1003.1-200x, <limits.h>

19358 **CHANGE HISTORY**

19359           First released in Issue 3.



19360 **Issue 4**

19361       Format reorganized.

19362       Internationalized environment variable support mandated.

19363 **Issue 6**

19364       The normative text is reworded to avoid use of the term “must” for application requirements.

## 19365 NAME

19366 get — get a version of an SCCS file (**DEVELOPMENT**)

## 19367 SYNOPSIS

19368 xSI get [-begkmlPst][-c *cutoff*][-i *list*][-r *SID*][-x *list*] *file...*

19369

## 19370 DESCRIPTION

19371 The *get* utility shall generate a text file from each named SCCS *file* according to the specifications  
19372 given by its options.19373 The generated text is normally written into a file called the **g-file** whose name is derived from  
19374 the SCCS file name by simply removing the leading **s**..

## 19375 OPTIONS

19376 The *get* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
19377 Section 12.2, Utility Syntax Guidelines.

19378 The following options shall be supported:

19379 **-r *SID*** Indicate the SCCS Identification String (SID) of the version (delta) of an SCCS file  
19380 to be retrieved. The table shows, for the most useful cases, what version of an  
19381 SCCS file is retrieved (as well as the SID of the version to be eventually created by  
19382 *delta* if the **-e** option is also used), as a function of the SID specified.19383 **-c *cutoff*** Indicate the *cutoff* date-time, in the form:

19384 YY[MM[DD[HH[MM[SS]]]]]

19385 For the YY component, values in the range [69-99] shall refer to years in the  
19386 twentieth century (1969 to 1999 inclusive); values in the range [00-68] shall refer to  
19387 years in the twenty-first century (2000 to 2068 inclusive).19388 No changes (deltas) to the SCCS file that were created after the specified *cutoff*  
19389 date-time are included in the generated text file. Units omitted from the date-time  
19390 default to their maximum possible values; for example, **-c 7502** is equivalent to **-c**  
19391 **750228235959**.19392 Any number of non-numeric characters may separate the various 2-digit pieces of  
19393 the *cutoff* date-time. This feature allows the user to specify a *cutoff* date in the form:  
19394 **-c "77/2/2 9:22:25"**.19395 **-e** Indicate that the *get* is for the purpose of editing or making a change (delta) to the  
19396 SCCS file via a subsequent use of *delta*. The **-e** option used in a *get* for a particular  
19397 version (SID) of the SCCS file prevents further *get* commands from editing on the  
19398 same SID until *delta* is executed or the **j** (joint edit) flag is set in the SCCS file.  
19399 Concurrent use of *get -e* for different SIDs is always allowed.19400 If the **g-file** generated by *get* with a **-e** option is accidentally ruined in the process  
19401 of editing, it may be regenerated by re-executing the *get* command with the **-k**  
19402 option in place of the **-e** option.19403 SCCS file protection specified via the ceiling, floor, and authorized user list stored  
19404 in the SCCS file is enforced when the **-e** option is used.19405 **-b** Use with the **-e** option to indicate that the new delta should have an SID in a new  
19406 branch as shown in the table below. This option is ignored if the **b** flag is not  
19407 present in the file or if the retrieved delta is not a leaf delta. (A leaf delta is one that  
19408 has no successors on the SCCS file tree.)

|       |                |                                                                                                                                                                                                                                                                     |
|-------|----------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 19409 |                | <b>Note:</b> A branch delta may always be created from a non-leaf delta.                                                                                                                                                                                            |
| 19410 | <b>-i list</b> | Indicate a <i>list</i> of deltas to be included (forced to be applied) in the creation of the generated file. The <i>list</i> has the following syntax:                                                                                                             |
| 19411 |                |                                                                                                                                                                                                                                                                     |
| 19412 |                | <list> ::= <range>   <list> , <range>                                                                                                                                                                                                                               |
| 19413 |                | <range> ::= SID   SID - SID                                                                                                                                                                                                                                         |
| 19414 |                | SID, the SCCS Identification of a delta, may be in any form shown in the "SID Specified" column of the table in the EXTENDED DESCRIPTION section. Partial SIDs are interpreted as shown in the "SID Retrieved" column of the table.                                 |
| 19415 |                |                                                                                                                                                                                                                                                                     |
| 19416 |                |                                                                                                                                                                                                                                                                     |
| 19417 | <b>-x list</b> | Indicate a <i>list</i> of deltas to be excluded (forced not to be applied) in the creation of the generated file. See the <b>-i</b> option for the <i>list</i> format.                                                                                              |
| 19418 |                |                                                                                                                                                                                                                                                                     |
| 19419 | <b>-k</b>      | Suppress replacement of identification keywords (see below) in the retrieved text by their value. The <b>-k</b> option is implied by the <b>-e</b> option.                                                                                                          |
| 19420 |                |                                                                                                                                                                                                                                                                     |
| 19421 | <b>-l</b>      | Write a delta summary into an <b>l-file</b> .                                                                                                                                                                                                                       |
| 19422 | <b>-L</b>      | Write a delta summary to standard output. All informative output that normally is written to standard output is written to standard error instead, unless the <b>-s</b> option is used, in which case it is suppressed.                                             |
| 19423 |                |                                                                                                                                                                                                                                                                     |
| 19424 |                |                                                                                                                                                                                                                                                                     |
| 19425 | <b>-p</b>      | Write the text retrieved from the SCCS file to the standard output. No <b>g-file</b> is created. All informative output that normally goes to the standard output goes to standard error instead, unless the <b>-s</b> option is used, in which case it disappears. |
| 19426 |                |                                                                                                                                                                                                                                                                     |
| 19427 |                |                                                                                                                                                                                                                                                                     |
| 19428 | <b>-s</b>      | Suppress all informative output normally written to standard output. However, fatal error messages (which are always written to the standard error) remain unaffected.                                                                                              |
| 19429 |                |                                                                                                                                                                                                                                                                     |
| 19430 |                |                                                                                                                                                                                                                                                                     |
| 19431 | <b>-m</b>      | Precede each text line retrieved from the SCCS file by the SID of the delta that inserted the text line in the SCCS file. The format is:                                                                                                                            |
| 19432 |                |                                                                                                                                                                                                                                                                     |
| 19433 |                | "%s\ts", <SID>, <text line>                                                                                                                                                                                                                                         |
| 19434 | <b>-n</b>      | Precede each generated text line with the %M% identification keyword value (see below). The format is:                                                                                                                                                              |
| 19435 |                |                                                                                                                                                                                                                                                                     |
| 19436 |                | "%s\ts", <%M% value>, <text line>                                                                                                                                                                                                                                   |
| 19437 |                | When both the <b>-m</b> and <b>-n</b> options are used, the <text line> is replaced by the <b>-m</b> option-generated format.                                                                                                                                       |
| 19438 |                |                                                                                                                                                                                                                                                                     |
| 19439 | <b>-g</b>      | Suppress the actual retrieval of text from the SCCS file. It is primarily used to generate an <b>l-file</b> , or to verify the existence of a particular SID.                                                                                                       |
| 19440 |                |                                                                                                                                                                                                                                                                     |
| 19441 | <b>-t</b>      | Use to access the most recently created (top) delta in a given release (for example, <b>-r 1</b> ), or release and level (for example, <b>-r 1.2</b> ).                                                                                                             |
| 19442 |                |                                                                                                                                                                                                                                                                     |

#### 19443 OPERANDS

19444 The following operands shall be supported:

|       |             |                                                                                                                                                                                                                                                                                                                    |
|-------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 19445 | <b>file</b> | A path name of an existing SCCS file or a directory. If <i>file</i> is a directory, <i>get</i> behaves as though each file in the directory were specified as a named file, except that non-SCCS files (last component of the path name does not begin with <b>s</b> .) and unreadable files are silently ignored. |
| 19446 |             |                                                                                                                                                                                                                                                                                                                    |
| 19447 |             |                                                                                                                                                                                                                                                                                                                    |
| 19448 |             |                                                                                                                                                                                                                                                                                                                    |
| 19449 |             | If a single instance <i>file</i> is specified as <b>'-'</b> , the standard input is read; each line of the standard input is taken to be the name of an SCCS file to be processed. Non-SCCS files and unreadable files are silently ignored.                                                                       |
| 19450 |             |                                                                                                                                                                                                                                                                                                                    |
| 19451 |             |                                                                                                                                                                                                                                                                                                                    |

19452 **STDIN**

19453       The standard input is a text file used only if the *file* operand is specified as *'-'*. Each line of the  
19454       text file is interpreted as an SCCS path name.

19455 **INPUT FILES**

19456       The SCCS files are files of an unspecified format.

19457 **ENVIRONMENT VARIABLES**

19458       The following environment variables shall affect the execution of *get*:

19459       *LANG*       Provide a default value for the internationalization variables that are unset or null.  
19460       If *LANG* is unset or null, the corresponding value from the implementation-  
19461       dependent default locale shall be used. If any of the internationalization variables  
19462       contains an invalid setting, the utility shall behave as if none of the variables had  
19463       been defined.

19464       *LC\_ALL*     If set to a non-empty string value, override the values of all the other  
19465       internationalization variables.

19466       *LC\_CTYPE*   Determine the locale for the interpretation of sequences of bytes of text data as  
19467       characters (for example, single-byte as opposed to multi-byte characters in  
19468       arguments and input files).

19469       *LC\_MESSAGES*

19470       Determine the locale that should be used to affect the format and contents of  
19471       diagnostic messages written to standard error, and informative messages written  
19472       to standard output (or standard error, if the *-p* option is used).

19473       *NLSPATH*    Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

19474 **ASYNCHRONOUS EVENTS**

19475       Default.

19476 **STDOUT**

19477       For each file processed, *get* shall write to standard output the SID being accessed and the number  
19478       of lines retrieved from the SCCS file, in the following format:

19479       "*%s\n%d lines\n*", *<SID>*, *<number of lines>*

19480       If the *-e* option is used, the SID of the delta to be made shall appear after the SID accessed and  
19481       before the number of lines generated, in the POSIX locale:

19482       "*%s\nnew delta %s\n%d\n*", *<SID accessed>*, *<SID to be made>*,  
19483       *<number of lines>*

19484       If there is more than one named file or if a directory or standard input is named, each path name  
19485       shall be written before each of the lines shown in one of the preceding formats:

19486       "*\n%s:\n*", *<pathname>*

19487       If the *-L* option is used, a delta summary shall be written following the format specified below  
19488       for **I-files**.

19489       If the *-i* option is used, included deltas are listed following the notation, in the POSIX locale:

19490       "*Included:\n*"

19491       If the *-x* option is used, excluded deltas are listed following the notation, in the POSIX locale:

19492       "*Excluded:\n*"

19493 If the **-p** or **-L** options are specified, the standard output consists of the text retrieved from the  
 19494 SCCS file.

#### 19495 **STDERR**

19496 The standard error shall be used only for diagnostic messages, except if the **-p** or **-L** options are  
 19497 specified, it includes all informative messages normally sent to standard output.

#### 19498 **OUTPUT FILES**

19499 Several auxiliary files may be created by *get*. These files are known generically as the **g-file**, **l-**  
 19500 **file**, **p-file**, and **z-file**. The letter before the hyphen is called the *tag*. An auxiliary file name is  
 19501 formed from the SCCS file name: the application shall ensure that the last component of all  
 19502 SCCS file names is of the form *s.module-name*; the auxiliary files are named by replacing the  
 19503 leading *s* with the tag. The **g-file** is an exception to this scheme: the **g-file** is named by removing  
 19504 the *s.* prefix. For example, for *s.xyz.c*, the auxiliary file names would be *xyz.c*, *l.xyz.c*, *p.xyz.c*,  
 19505 and *z.xyz.c*, respectively.

19506 The **g-file**, which contains the generated text, is created in the current directory (unless the **-p**  
 19507 option is used). A **g-file** is created in all cases, whether or not any lines of text were generated by  
 19508 the *get*. It is owned by the real user. If the **-k** option is used or implied, it is writable by the  
 19509 owner only (read-only for everyone else); otherwise, it is read-only. Only the real user need have  
 19510 write permission in the current directory.

19511 The **l-file** contains a table showing which deltas were applied in generating the retrieved text.  
 19512 The **l-file** is created in the current directory if the **-l** option is used; it is read-only and it is  
 19513 owned by the real user. Only the real user need have write permission in the current directory.

19514 Lines in the **l-file** have the following format:

19515 "%c%c%cΔ%s\t%sΔ%s\n", <code1>, <code2>, <code3>,  
 19516 <SID>, <date-time>, <login>

19517 where the entries are:

19518 <code1> A <space> character if the delta was applied; ' \* ' otherwise.  
 19519 <code2> A <space> character if the delta was applied or was not applied and ignored; ' \* '  
 19520 if the delta was not applied and was not ignored.  
 19521 <code3> A character indicating a special reason why the delta was or was not applied:  
 19522 **I** Included.  
 19523 **X** Excluded.  
 19524 **C** Cut off (by a **-c** option).  
 19525 <date-time> Date and time (using the *date* utility's %y/%m/%d %T format) of creation.  
 19526 <login> Login name of person who created *delta*.

19527 The comments and MR data follow on subsequent lines, indented one <tab> character. A blank  
 19528 line terminates each entry.

19529 The **p-file** is used to pass information resulting from a *get* with a **-e** option along to *delta*. Its  
 19530 contents are also used to prevent a subsequent execution of *get* with a **-e** option for the same SID  
 19531 until *delta* is executed or the joint edit flag, **j**, is set in the SCCS file. The **p-file** is created in the  
 19532 directory containing the SCCS file and the application shall ensure that the effective user has  
 19533 write permission in that directory. It is writable by owner only, and it is owned by the effective  
 19534 user. Each line in the **p-file** has the following format:

19535 "%sΔ%sΔ%sΔ%sΔ%sπs%s\n", <g-file SID>, <SID of new delta>,  
 19536 <login-name of real user>, <date-time>, <i-value>, <x-value>

## 19537 **Notes to Reviewers**

19538 *This section with side shading will not appear in the final copy. - Ed.*

19539 The example above has been fixed to include  $\pi$ .

19540 where <i-value> is the value of the *list* option-argument to **-i** (or null) and <x-value> is the value  
 19541 of the *list* option-argument to **-x** (or null). There can be an arbitrary number of lines in the **p-file**  
 19542 at any time; no two lines can have the same new delta SID.

19543 The **z-file** serves as a lock-out mechanism against simultaneous updates. Its contents are the  
 19544 binary process ID of the command (that is, *get*) that created it. The **z-file** is created in the  
 19545 directory containing the SCCS file for the duration of *get*. The same protection restrictions as  
 19546 those for the **p-file** apply for the **z-file**. The **z-file** is created read-only.

## 19547 **EXTENDED DESCRIPTION**

| Determination of SCCS Identification String |                       |                                                |                  |                               |
|---------------------------------------------|-----------------------|------------------------------------------------|------------------|-------------------------------|
| SID*<br>Specified                           | -b Keyletter<br>Used† | Other<br>Conditions                            | SID<br>Retrieved | SID of Delta<br>to be Created |
| none‡                                       | no                    | R defaults to mR                               | mR.mL            | mR.(mL+1)                     |
| none‡                                       | yes                   | R defaults to mR                               | mR.mL            | mR.mL.(mB+1).1                |
| R                                           | no                    | R > mR                                         | mR.mL            | R.1***                        |
| R                                           | no                    | R = mR                                         | mR.mL            | mR.(mL+1)                     |
| R                                           | yes                   | R > mR                                         | mR.mL            | mR.mL.(mB+1).1                |
| R                                           | yes                   | R = mR                                         | mR.mL            | mR.mL.(mB+1).1                |
| R                                           | –                     | R < mR and<br>R does not exist                 | hR.mL**          | hR.mL.(mB+1).1                |
| R                                           | –                     | Trunk successor in release > R<br>and R exists | R.mL             | R.mL.(mB+1).1                 |
| R.L                                         | no                    | No trunk successor                             | R.L              | R.(L+1)                       |
| R.L                                         | yes                   | No trunk successor                             | R.L              | R.L.(mB+1).1                  |
| R.L                                         | –                     | Trunk successor<br>in release ≥ R              | R.L              | R.L.(mB+1).1                  |
| R.L.B                                       | no                    | No branch successor                            | R.L.B.mS         | R.L.B.(mS+1)                  |
| R.L.B                                       | yes                   | No branch successor                            | R.L.B.mS         | R.L.(mB+1).1                  |
| R.L.B.S                                     | no                    | No branch successor                            | R.L.B.S          | R.L.B.(S+1)                   |
| R.L.B.S                                     | yes                   | No branch successor                            | R.L.B.S          | R.L.(mB+1).1                  |
| R.L.B.S                                     | –                     | Branch successor                               | R.L.B.S          | R.L.(mB+1).1                  |

19570 \* R, L, B, and S are the release, level, branch, and sequence components of the SID,  
 19571 respectively; m means maximum. Thus, for example, R.mL means “the maximum level  
 19572 number within release R”; R.L.(mB+1).1 means “the first sequence number on the new  
 19573 branch (that is, maximum branch number plus one) of level L within release R”. Note  
 19574 that if the SID specified is of the form R.L, R.L.B, or R.L.B.S, each of the specified  
 19575 components shall exist.

19576 \*\* hR is the highest existing release that is lower than the specified, nonexistent, release R.

|       |                                                                                             |                                                                                                                                                                                                                                                                                                |
|-------|---------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 19577 | ***                                                                                         | This is used to force creation of the first delta in a new release.                                                                                                                                                                                                                            |
| 19578 | †                                                                                           | The <b>–b</b> option is effective only if the <b>b</b> flag is present in the file. An entry of '–' means “irrelevant”.                                                                                                                                                                        |
| 19579 |                                                                                             |                                                                                                                                                                                                                                                                                                |
| 19580 | ‡                                                                                           | This case applies if the <b>d</b> (default SID) flag is not present in the file. If the <b>d</b> flag is present in the file, then the SID obtained from the <b>d</b> flag is interpreted as if it had been specified on the command line. Thus, one of the other cases in this table applies. |
| 19581 |                                                                                             |                                                                                                                                                                                                                                                                                                |
| 19582 |                                                                                             |                                                                                                                                                                                                                                                                                                |
|       |                                                                                             |                                                                                                                                                                                                                                                                                                |
| 19583 | <b>Identification Keywords</b>                                                              |                                                                                                                                                                                                                                                                                                |
| 19584 | Identifying information is inserted into the text retrieved from the SCCS file by replacing |                                                                                                                                                                                                                                                                                                |
| 19585 | identification keywords with their value wherever they occur. The following keywords may be |                                                                                                                                                                                                                                                                                                |
| 19586 | used in the text stored in an SCCS file:                                                    |                                                                                                                                                                                                                                                                                                |
| 19587 | %M%                                                                                         | Module name: either the value of the <b>m</b> flag in the file, or if absent, the name of the SCCS file with the leading <b>s.</b> removed.                                                                                                                                                    |
| 19588 |                                                                                             |                                                                                                                                                                                                                                                                                                |
| 19589 | %I%                                                                                         | SCCS identification (SID) (%R%.%L% or %R%.%L%.%B%.%S%) of the retrieved text.                                                                                                                                                                                                                  |
| 19590 |                                                                                             |                                                                                                                                                                                                                                                                                                |
| 19591 | %R%                                                                                         | Release.                                                                                                                                                                                                                                                                                       |
| 19592 | %L%                                                                                         | Level.                                                                                                                                                                                                                                                                                         |
| 19593 | %B%                                                                                         | Branch.                                                                                                                                                                                                                                                                                        |
| 19594 | %S%                                                                                         | Sequence.                                                                                                                                                                                                                                                                                      |
| 19595 | %D%                                                                                         | Current date (YY/MM/DD).                                                                                                                                                                                                                                                                       |
| 19596 | %H%                                                                                         | Current date (MM/DD/YY).                                                                                                                                                                                                                                                                       |
| 19597 | %T%                                                                                         | Current time (HH:MM:SS).                                                                                                                                                                                                                                                                       |
| 19598 | %E%                                                                                         | Date newest applied delta was created (YY/MM/DD).                                                                                                                                                                                                                                              |
| 19599 | %G%                                                                                         | Date newest applied delta was created (MM/DD/YY).                                                                                                                                                                                                                                              |
| 19600 | %U%                                                                                         | Time newest applied delta was created (HH:MM:SS).                                                                                                                                                                                                                                              |
| 19601 | %Y%                                                                                         | Module type: value of the <b>t</b> flag in the SCCS file.                                                                                                                                                                                                                                      |
| 19602 | %F%                                                                                         | SCCS file name.                                                                                                                                                                                                                                                                                |
| 19603 | %P%                                                                                         | SCCS absolute path name.                                                                                                                                                                                                                                                                       |
| 19604 | %Q%                                                                                         | The value of the <b>q</b> flag in the file.                                                                                                                                                                                                                                                    |
| 19605 | %C%                                                                                         | Current line number. This keyword is intended for identifying messages output by the program, such as “this should not have happened” type errors. It is not intended to be used on every line to provide sequence numbers.                                                                    |
| 19606 |                                                                                             |                                                                                                                                                                                                                                                                                                |
| 19607 |                                                                                             |                                                                                                                                                                                                                                                                                                |
| 19608 | %Z%                                                                                         | The four-character string "@( # )" recognizable by <i>what</i> .                                                                                                                                                                                                                               |
| 19609 | %W%                                                                                         | A shorthand notation for constructing <i>what</i> strings:                                                                                                                                                                                                                                     |
| 19610 |                                                                                             |                                                                                                                                                                                                                                                                                                |
| 19611 | %A%                                                                                         | Another shorthand notation for constructing <i>what</i> strings:                                                                                                                                                                                                                               |
| 19612 |                                                                                             |                                                                                                                                                                                                                                                                                                |

19613 **EXIT STATUS**

19614           The following exit values shall be returned:

19615           0   Successful completion.

19616           &gt;0   An error occurred.

19617 **CONSEQUENCES OF ERRORS**

19618           Default.

19619 **APPLICATION USAGE**

19620           None.

19621 **EXAMPLES**

19622           None.

19623 **RATIONALE**

19624           None.

19625 **FUTURE DIRECTIONS**19626           The **-lp** option may be withdrawn in a future issue.19627 **SEE ALSO**19628           *admin, delta, prs, what*19629 **CHANGE HISTORY**

19630           First released in Issue 2.

19631 **Issue 4**

19632           Format reorganized.

19633           Exceptions to Utility Syntax Guidelines conformance noted.

19634           Internationalized environment variable support mandated.

19635 **Issue 5**

19636           Correction to the first format string in STDOUT.

19637           The interpretation of the YY component of the **-c cutoff** argument is noted.19638 **Issue 6**19639           The obsolescent SYNOPSIS is removed, removing the **-lp** option.19640           The Open Group corrigenda item U025/5 has been applied, correcting text in the OPTIONS  
19641           section. |

19642           The normative text is reworded to avoid use of the term “must” for application requirements. |



19643 **NAME**

19644           getconf — get configuration values

19645 **SYNOPSIS**19646 MAN       getconf [ *-v specification* ] *system\_var*19647 MAN       getconf [ *-v specification* ] *path\_var pathname*19648 **DESCRIPTION**

19649       In the first synopsis form, the *getconf* utility shall write to the standard output the value of the  
 19650       variable specified by the *system\_var* operand.

19651       In the second synopsis form, the *getconf* utility shall write to the standard output the value of the  
 19652       variable specified by the *path\_var* operand for the path specified by the *pathname* operand.

19653       The value of each configuration variable shall be determined as if it were obtained by calling the  
 19654       function from which it is defined to be available by this volume of IEEE Std. 1003.1-200x or by  
 19655       the System Interfaces volume of IEEE Std. 1003.1-200x (see the OPERANDS section). The value  
 19656       shall reflect conditions in the current operating environment.

19657 **OPTIONS**

19658       The *getconf* utility shall conform to the System Interface Definitions volume of  
 19659       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

19660       The following option shall be supported:

19661       *-v specification*

19662 MAN       Indicate a specific specification and version for which configuration variables shall  
 19663       be determined. If this option is not specified, the values returned correspond to an  
 19664       implementation default conforming compilation environment.

19665       If the command:

19666       getconf \_XBS5\_ILP32\_OFF32

19667       does not write "-1\n" or "undefined\n" to standard output, then commands of  
 19668       the form:

19669       getconf -v XBS5\_ILP32\_OFF32 ...

19670       determine values for configuration variables corresponding to the  
 19671       XBS5\_ILP32\_OFF32 compilation environment specified in c89 on page 246,  
 19672       EXTENDED DESCRIPTION.

19673       If the command:

19674       getconf \_XBS5\_ILP32\_OFFBIG

19675       does not write "-1\n" or "undefined\n" to standard output, then commands of  
 19676       the form:

19677       getconf -v XBS5\_ILP32\_OFFBIG ...

19678       determine values for configuration variables corresponding to the  
 19679       XBS5\_ILP32\_OFFBIG compilation environment specified in c89 on page 246,  
 19680       EXTENDED DESCRIPTION.

19681       If the command:

19682       getconf \_XBS5\_LP64\_OFF64

19683       does not write "-1\n" or "undefined\n" to standard output, then commands of  
 19684       the form:

19685           getconf -v XBS5\_LP64\_OFF64 ...

19686           determine values for configuration variables corresponding to the  
 19687           XBS5\_LP64\_OFF64 compilation environment specified in *c89* on page 246,  
 19688           EXTENDED DESCRIPTION.

19689           If the command:

19690           getconf \_XBS5\_LPBIG\_OFFBIG

19691           does not write "-1\n" or "undefined\n" to standard output, then commands of  
 19692           the form:

19693           getconf -v XBS5\_LPBIG\_OFFBIG ...

19694           determine values for configuration variables corresponding to the  
 19695           XBS5\_LPBIG\_OFFBIG compilation environment specified in *c89* on page 246,  
 19696           EXTENDED DESCRIPTION.

#### 19697 OPERANDS

19698           The following operands shall be supported:

|       |                   |                                                                                                                                                                                                                                                        |
|-------|-------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 19699 | <i>path_var</i>   | A name of a configuration variable. All of the variables in the <i>pathconf()</i> function defined in the System Interfaces volume of IEEE Std. 1003.1-200x are supported and the implementation may add other local variables.                        |
| 19700 |                   |                                                                                                                                                                                                                                                        |
| 19701 |                   |                                                                                                                                                                                                                                                        |
| 19702 | <i>pathname</i>   | A path name for which the variable specified by <i>path_var</i> is to be determined.                                                                                                                                                                   |
| 19703 | <i>system_var</i> | A name of a configuration variable. All of the variables in the <i>confstr()</i> and <i>sysconf()</i> functions defined in the System Interfaces volume of IEEE Std. 1003.1-200x shall be supported and the implementation may add other local values. |
| 19704 |                   |                                                                                                                                                                                                                                                        |
| 19705 |                   |                                                                                                                                                                                                                                                        |
| 19706 |                   |                                                                                                                                                                                                                                                        |

#### 19707 Notes to Reviewers

19708           *This section with side shading will not appear in the final copy. - Ed.*

19709           D1, XCU, ERN 256 notes an issue - we may change the names of these macros for  
 19710           the next revision.

|       |     |                                                                                                          |
|-------|-----|----------------------------------------------------------------------------------------------------------|
| 19711 | MAN | When the symbol listed in the first column of the following table is used as the                         |
| 19712 |     | <i>system_var</i> operand, <i>getconf</i> yields the same value as <i>confstr()</i> when called with the |
| 19713 |     | value in the second column:                                                                              |

19714

19715

19716

19717

19718

19719

19720

19721

19722

19723

19724

19725

19726

19727

19728

19729

19730

19731

19732

|  | <i>system_var</i>           | <i>confstr()</i> Name Value     |
|--|-----------------------------|---------------------------------|
|  | PATH                        | _CS_PATH                        |
|  | XBS5_ILP32_OFF32_CFLAGS     | _CS_XBS5_ILP32_OFF32_CFLAGS     |
|  | XBS5_ILP32_OFF32_LDFLAGS    | _CS_XBS5_ILP32_OFF32_LDFLAGS    |
|  | XBS5_ILP32_OFF32_LIBS       | _CS_XBS5_ILP32_OFF32_LIBS       |
|  | XBS5_ILP32_OFF32_LINTFLAGS  | _CS_XBS5_ILP32_OFF32_LINTFLAGS  |
|  | XBS5_ILP32_OFFBIG_CFLAGS    | _CS_XBS5_ILP32_OFFBIG_CFLAGS    |
|  | XBS5_ILP32_OFFBIG_LDFLAGS   | _CS_XBS5_ILP32_OFFBIG_LDFLAGS   |
|  | XBS5_ILP32_OFFBIG_LIBS      | _CS_XBS5_ILP32_OFFBIG_LIBS      |
|  | XBS5_ILP32_OFFBIG_LINTFLAGS | _CS_XBS5_ILP32_OFFBIG_LINTFLAGS |
|  | XBS5_LP64_OFF64_CFLAGS      | _CS_XBS5_LP64_OFF64_CFLAGS      |
|  | XBS5_LP64_OFF64_LDFLAGS     | _CS_XBS5_LP64_OFF64_LDFLAGS     |
|  | XBS5_LP64_OFF64_LIBS        | _CS_XBS5_LP64_OFF64_LIBS        |
|  | XBS5_LP64_OFF64_LINTFLAGS   | _CS_XBS5_LP64_OFF64_LINTFLAGS   |
|  | XBS5_LPBIG_OFFBIG_CFLAGS    | _CS_XBS5_LPBIG_OFFBIG_CFLAGS    |
|  | XBS5_LPBIG_OFFBIG_LDFLAGS   | _CS_XBS5_LPBIG_OFFBIG_LDFLAGS   |
|  | XBS5_LPBIG_OFFBIG_LIBS      | _CS_XBS5_LPBIG_OFFBIG_LIBS      |
|  | XBS5_LPBIG_OFFBIG_LINTFLAGS | _CS_XBS5_LPBIG_OFFBIG_LINTFLAGS |

19733 **STDIN**

19734 Not used.

19735 **INPUT FILES**

19736 None.

19737 **ENVIRONMENT VARIABLES**19738 The following environment variables shall affect the execution of *getconf*:

19739 **LANG** Provide a default value for the internationalization variables that are unset or null.  
 19740 If *LANG* is unset or null, the corresponding value from the implementation-  
 19741 dependent default locale shall be used. If any of the internationalization variables  
 19742 contains an invalid setting, the utility shall behave as if none of the variables had  
 19743 been defined.

19744 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 19745 internationalization variables.

19746 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 19747 characters (for example, single-byte as opposed to multi-byte characters in  
 19748 arguments).

19749 **LC\_MESSAGES**  
 19750 Determine the locale that should be used to affect the format and contents of  
 19751 diagnostic messages written to standard error.

19752 **XS1 NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

19753 **ASYNCHRONOUS EVENTS**

19754 Default.

19755 **STDOUT**

19756 If the specified variable is defined on the system and its value is described to be available from  
 19757 the *confstr()* function defined in the System Interfaces volume of IEEE Std. 1003.1-200x, its value  
 19758 shall be written in the following format:

19759 "%s\n", &lt;value&gt;

19760 Otherwise, if the specified variable is defined on the system, its value shall be written in the  
 19761 following format:

19762 "%d\n", <value>

19763 If the specified variable is valid, but is undefined on the system, *getconf* shall write using the  
 19764 following format:

19765 "undefined\n"

19766 If the variable name is invalid or an error occurs, nothing shall be written to standard output.

#### 19767 **STDERR**

19768 Used only for diagnostic messages.

#### 19769 **OUTPUT FILES**

19770 None.

#### 19771 **EXTENDED DESCRIPTION**

19772 None.

#### 19773 **EXIT STATUS**

19774 The following exit values shall be returned:

19775 0 The specified variable is valid and information about its current state was written  
 19776 successfully.

19777 >0 An error occurred.

#### 19778 **CONSEQUENCES OF ERRORS**

19779 Default.

#### 19780 **APPLICATION USAGE**

19781 None.

#### 19782 **EXAMPLES**

19783 The following example illustrates the value of {NGROUPS\_MAX}:

19784 `getconf NGROUPS_MAX`

19785 The following example illustrates the value of {NAME\_MAX} for a specific directory:

19786 `getconf NAME_MAX /usr`

19787 The following example shows how to deal more carefully with results that might be unspecified:

```
19788 if value=$(getconf PATH_MAX /usr); then
19789 if ["$value" = "undefined"]; then
19790 echo PATH_MAX in /usr is infinite.
19791 else
19792 echo PATH_MAX in /usr is $value.
19793 fi
19794 else
19795 echo Error in getconf.
19796 fi
```

19797 Note that:

19798 `sysconf(_SC_POSIX_C_BIND);`

19799 and:

19800 `system("getconf POSIX2_C_BIND");`

19801 in a C program could give different answers. The *sysconf()* call supplies a value that corresponds  
 19802 to the conditions when the program was either compiled or executed, depending on the  
 19803 implementation; the *system()* call to *getconf* always supplies a value corresponding to conditions  
 19804 when the program is executed.

#### 19805 **RATIONALE**

19806 The original need for this utility, and for the *confstr()* function, was to provide a way of finding  
 19807 the configuration-defined default value for the *PATH* environment variable. Since *PATH* can be  
 19808 modified by the user to include directories that could contain utilities replacing the standard  
 19809 utilities, shell scripts need a way to determine the system-supplied *PATH* environment variable  
 19810 value that contains the correct search path for the standard utilities. It was later suggested that  
 19811 access to the other variables described in this volume of IEEE Std. 1003.1-200x could also be  
 19812 useful to applications.

19813 The following two function calls in a C program could give different answers:

19814 `sysconf(_SC_POSIX_C_BIND);`  
 19815 `system("getconf POSIX2_C_BIND");`

19816 The *sysconf()* call supplies a value that corresponds to the conditions when the program was  
 19817 either compiled or executed, depending on the implementation; the *system()* call to *getconf*  
 19818 always supplies a value corresponding to conditions when the program is executed.

19819 This functionality of *getconf* would not be adequately subsumed by another command such as:

19820 `grep var /etc/conf`

19821 because such a strategy would provide correct values for neither those variables that can vary at  
 19822 runtime, nor those that can vary depending on the path.

19823 Early proposal versions of *getconf* specified exit status 1 when the specified variable was valid,  
 19824 but not defined on the system. The output string "undefined" is now used to specify this case  
 19825 with exit code 0 because so many things depend on an exit code of zero when an invoked utility  
 19826 is successful.

#### 19827 **FUTURE DIRECTIONS**

19828 None.

#### 19829 **SEE ALSO**

19830 *c89* the System Interfaces volume of IEEE Std. 1003.1-200x, *confstr()*, *pathconf()*, *sysconf()*

#### 19831 **CHANGE HISTORY**

19832 First released in Issue 4.

#### 19833 **Issue 4, Version 2**

19834 The following changes are made in the table of values for *system\_var*:

- 19835 • Names beginning with *POSIX\_* are changed to begin with *\_POSIX\_*.
- 19836 • Names beginning with *XOPEN\_* are changed to begin with *\_XOPEN\_*.
- 19837 • {*MN\_NMAX*} is changed to {*NL\_MAX*}.
- 19838 • {*NL\_SET\_MAX*} is changed to {*NL\_SETMAX*}.
- 19839 • {*NL\_TEXT\_MAX*} is changed to {*NL\_TEXTMAX*}.
- 19840 • The *\_XOPEN\_CRYPT*, *\_XOPEN\_ENH\_I18N*, and *\_XOPEN\_SHM* configuration variables are  
 19841 added to the list.

19842 **Issue 5**

19843 In the OPERANDS section:

- 19844 • {NL\_MAX} is changed to {NL\_NMAX}.
- 19845 • Entries beginning NL\_ are deleted from the list of standard configuration variables.
- 19846 • The list of variables previously marked UX is merged with the list marked EX.
- 19847 • Operands are added to support new Option Groups.
- 19848 • Operands are added so that *getconf* can determine supported programming environments.

19849 **Issue 6**19850 The Open Group corrigenda item U029/4 has been applied, correcting the example command in  
19851 the last paragraph of the OPTIONS section.19852 The following new requirements on POSIX implementations derive from alignment with the  
19853 Single UNIX Specification:

- 19854 • Operands are added to determine supported programming environments.

19855 **NAME**

19856       getopts — parse utility options

19857 **SYNOPSIS**19858       getopts *optstring name* [*arg...*]19859 **DESCRIPTION**

19860       The *getopts* utility can be used to retrieve options and option-arguments from a list of  
 19861       parameters. It shall support the Utility Syntax Guidelines 3 to 10, inclusive, described in the  
 19862       System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax  
 19863       Guidelines.

19864       Each time it is invoked, the *getopts* utility shall place the value of the next option in the shell  
 19865       variable specified by the *name* operand and the index of the next argument to be processed in the  
 19866       shell variable *OPTIND*. Whenever the shell is invoked, *OPTIND* shall be initialized to 1.

19867       When the option requires an option-argument, the *getopts* utility shall place it in the shell  
 19868       variable *OPTARG*. If no option was found, or if the option that was found does not have an  
 19869       option-argument, *OPTARG* shall be unset.

19870       If an option character not contained in the *optstring* operand is found where an option character  
 19871       is expected, the shell variable specified by *name* shall be set to the question-mark ('?') character.  
 19872       In this case, if the first character in *optstring* is a colon (':'), the shell variable *OPTARG* shall be  
 19873       set to the option character found, but no output shall be written to standard error; otherwise, the  
 19874       shell variable *OPTARG* shall be unset and a diagnostic message shall be written to standard  
 19875       error. This condition shall be considered to be an error detected in the way arguments were  
 19876       presented to the invoking application, but shall be not an error in *getopts* processing.

19877       If an option-argument is missing:

- 19878       • If the first character of *optstring* is a colon, the shell variable specified by *name* shall be set to  
 19879       the colon character and the shell variable *OPTARG* shall be set to the option character found.
- 19880       • Otherwise, the shell variable specified by *name* shall be set to the question-mark character,  
 19881       the shell variable *OPTARG* shall be unset, and a diagnostic message shall be written to  
 19882       standard error. This condition shall be considered to be an error detected in the way  
 19883       arguments were presented to the invoking application, but shall not be an error in *getopts*  
 19884       processing; a diagnostic message shall be written as stated, but the exit status shall be zero.

19885       When the end of options is encountered, the *getopts* utility shall exit with a return value greater  
 19886       than zero; the shell variable *OPTIND* shall be set to the index of the first non-option-argument,  
 19887       where the first "—" argument is considered to be an option-argument if there are no other non-  
 19888       option-arguments appearing before it, or the value "\$#+1" if there are no non-option-  
 19889       arguments; the *name* variable shall be set to the question-mark character. Any of the following  
 19890       shall identify the end of options: the special option "—", finding an argument that does not  
 19891       begin with a '—', or encountering an error.

19892       The shell variables *OPTIND* and *OPTARG* shall be local to the caller of *getopts* and shall not be  
 19893       exported by default.

19894       The shell variable specified by the *name* operand, *OPTIND* and *OPTARG* shall affect the current  
 19895       shell execution environment; see Section 2.12 on page 90.

19896       If the application sets *OPTIND* to the value 1, a new set of parameters can be used: either the  
 19897       current positional parameters or new *arg* values. Any other attempt to invoke *getopts* multiple  
 19898       times in a single shell execution environment with parameters (positional parameters or *arg*  
 19899       operands) that are not the same in all invocations, or with an *OPTIND* value modified to be a  
 19900       value other than 1, produces unspecified results.

19901 **OPTIONS**

19902       None.

19903 **OPERANDS**

19904       The following operands shall be supported:

19905       *optstring*     A string containing the option characters recognized by the utility invoking *getopts*.  
 19906                       If a character is followed by a colon, the option shall be expected to have an  
 19907                       argument, which should be supplied as a separate argument. Applications should  
 19908                       specify an option character and its option-argument as separate arguments, but  
 19909                       *getopts* shall interpret the characters following an option character requiring  
 19910                       arguments as an argument whether or not this is done. An explicit null option-  
 19911                       argument need not be recognized if it is not supplied as a separate argument when  
 19912                       *getopts* is invoked. (See also the *getopt()* function defined in the System Interfaces  
 19913                       volume of IEEE Std. 1003.1-200x.) The characters question-mark and colon shall  
 19914                       not be used as option characters by an application. The use of other option  
 19915                       characters that are not alphanumeric produces unspecified results. If the option-  
 19916                       argument is not supplied as a separate argument from the option character, the  
 19917                       value in *OPTARG* shall be stripped of the option character and the '-'. The first  
 19918                       character in *optstring* determines how *getopts* behaves if an option character is not  
 19919                       known or an option-argument is missing.

19920       *name*         The name of a shell variable that shall be set by the *getopts* utility to the option  
 19921                       character that was found.

19922       The *getopts* utility by default shall parse positional parameters passed to the invoking shell  
 19923       procedure. If *args* are given, they shall be parsed instead of the positional parameters.

19924 **STDIN**

19925       Not used.

19926 **INPUT FILES**

19927       None.

19928 **ENVIRONMENT VARIABLES**19929       The following environment variables shall affect the execution of *getopts*:

19930       *LANG*         Provide a default value for the internationalization variables that are unset or null.  
 19931                       If *LANG* is unset or null, the corresponding value from the implementation-  
 19932                       dependent default locale shall be used. If any of the internationalization variables  
 19933                       contains an invalid setting, the utility shall behave as if none of the variables had  
 19934                       been defined.

19935       *LC\_ALL*       If set to a non-empty string value, override the values of all the other  
 19936                       internationalization variables.

19937       *LC\_CTYPE*     Determine the locale for the interpretation of sequences of bytes of text data as  
 19938                       characters (for example, single-byte as opposed to multi-byte characters in  
 19939                       arguments and input files).

19940       *LC\_MESSAGES*

19941                       Determine the locale that should be used to affect the format and contents of  
 19942                       diagnostic messages written to standard error.

19943 XSI     *NLSPATH*   Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

19944       *OPTIND*       This variable shall be used by the *getopts* utility as the index of the next argument  
 19945                       to be processed.



19946 **ASYNCHRONOUS EVENTS**

19947 Default.

19948 **STDOUT**

19949 Not used.

19950 **STDERR**

19951 Whenever an error is detected and the first character in the *optstring* operand is not a colon  
 19952 (' : '), a diagnostic message shall be written to standard error with the following information in  
 19953 an unspecified format:

- 19954 • The invoking program name shall be identified in the message. The invoking program name  
 19955 shall be the value of the shell special parameter 0 (see Section 2.5.2 on page 43) at the time the  
 19956 *getopts* utility is invoked. A name equivalent to:

19957 `basename "$0"`

19958 may be used.

- 19959 • If an option is found that was not specified in *optstring*, this error is identified and the invalid  
 19960 option character shall be identified in the message.

- 19961 • If an option requiring an option-argument is found, but an option-argument is not found,  
 19962 this error shall be identified and the invalid option character shall be identified in the  
 19963 message.

19964 **OUTPUT FILES**

19965 None.

19966 **EXTENDED DESCRIPTION**

19967 None.

19968 **EXIT STATUS**

19969 The following exit values shall be returned:

19970 `0` An option, specified or unspecified by *optstring*, was found.19971 `>0` The end of options was encountered or an error occurred.19972 **CONSEQUENCES OF ERRORS**

19973 Default.

19974 **APPLICATION USAGE**

19975 Since *getopts* affects the current shell execution environment, it is generally provided as a shell  
 19976 regular built-in. If it is called in a subshell or separate utility execution environment, such as one  
 19977 of the following:

19978 `(getopts abc value "$@")`19979 `nohup getopts ...`19980 `find . -exec getopts ... \;`

19981 it does not affect the shell variables in the caller's environment.

19982 Note that shell functions share *OPTIND* with the calling shell even though the positional  
 19983 parameters are changed. If the calling shell and any of its functions uses *getopts* to parse  
 19984 arguments, the results are unspecified.

19985 **EXAMPLES**

19986 The following example script parses and displays its arguments:

19987 `aflag=`19988 `bflag=`

```

19989 while getopts ab: name
19990 do
19991 case $name in
19992 a) aflag=1;;
19993 b) bflag=1
19994 bval="$OPTARG";;
19995 ?) printf "Usage: %s: [-a] [-b value] args\n" $0
19996 exit 2;;
19997 esac
19998 done
19999 if [! -z "$aflag"]; then
20000 printf "Option -a specified\n"
20001 fi
20002 if [! -z "$bflag"]; then
20003 printf 'Option -b "%s" specified\n' "$bval"
20004 fi
20005 shift $(($OPTIND - 1))
20006 printf "Remaining arguments are: %s\n" "$*"

```

#### 20007 RATIONALE

20008 The *getopts* utility was chosen in preference to the System V *getopt* utility because *getopts* handles  
 20009 option-arguments containing <blank> characters.

20010 The *OPTARG* variable is not mentioned in the ENVIRONMENT VARIABLES section because it  
 20011 does not affect the execution of *getopts*; it is one of the few “output-only” variables used by the  
 20012 standard utilities.

20013 The colon is not allowed as an option character because that is not historical behavior, and it  
 20014 violates the Utility Syntax Guidelines. The colon is now specified to behave as in the KornShell  
 20015 version of the *getopts* utility; when used as the first character in the *optstring* operand, it disables  
 20016 diagnostics concerning missing option-arguments and unexpected option characters. This  
 20017 replaces the use of the *OPTERR* variable that was specified in an early proposal.

20018 The formats of the diagnostic messages produced by the *getopts* utility and the *getopt()* function  
 20019 are not fully specified because implementations with superior (“friendlier”) formats objected to  
 20020 the formats used by some historical implementations. The standard developers considered it  
 20021 important that the information in the messages used be uniform between *getopts* and *getopt()*.  
 20022 Exact duplication of the messages might not be possible, particularly if a utility is built on  
 20023 another system that has a different *getopt()* function, but the messages must have specific  
 20024 information included so that the program name, invalid option character, and type of error can  
 20025 be distinguished by a user.

20026 Only a rare application program intercepts a *getopts* standard error message and wants to parse  
 20027 it. Therefore, implementations are free to choose the most usable messages they can devise. The  
 20028 following formats are used by many historical implementations:

```

20029 "%s: illegal option -- %c\n", <program name>, <option character>
20030 "%s: option requires an argument -- %c\n", <program name>, \
20031 <option character>

```

20032 Historical shells with built-in versions of *getopt()* or *getopts* have used different formats,  
 20033 frequently not even indicating the option character found in error.

20034 **FUTURE DIRECTIONS**

20035 None.

20036 **SEE ALSO**20037 The System Interfaces volume of IEEE Std. 1003.1-200x, *getopt()*20038 **CHANGE HISTORY**

20039 First released in Issue 4.

20040 **Issue 6**

20041 The normative text is reworded to avoid use of the term “must” for application requirements.

## 20042 NAME

20043       grep — search a file for a pattern

## 20044 SYNOPSIS

20045       grep [-E | -F][-c | -l | -q][-insvx] -e *pattern\_list*...  
20046       [-f *pattern\_file*]...[*file*...]20047       grep [-E | -F][-c | -l | -q][-insvx][-e *pattern\_list*]...  
20048       -f *pattern\_file*...[*file*...]20049       grep [-E | -F][-c | -l | -q][-insvx] *pattern\_list*[*file*...]

## 20050 DESCRIPTION

20051       The *grep* utility shall search the input files, selecting lines matching one or more patterns; the  
 20052       types of patterns are controlled by the options specified. The patterns are specified by the **-e**  
 20053       option, **-f** option, or the *pattern\_list* operand. The *pattern\_list*'s value shall consist of one or more  
 20054       patterns separated by <newline> characters; the *pattern\_file*'s contents shall consist of one or  
 20055       more patterns terminated by <newline> characters. By default, an input line shall be selected if  
 20056       any pattern, treated as an entire basic regular expression (BRE) as described in the System  
 20057       Interface Definitions volume of IEEE Std. 1003.1-200x, Section 9.3, Basic Regular Expressions,  
 20058       matches any part of the line; a null BRE shall match every line. By default, each selected input  
 20059       line shall be written to the standard output.

20060       Regular expression matching shall be based on text lines. Since a <newline> character separates  
 20061       or terminates patterns (see the **-e** and **-f** options below), regular expressions cannot contain a  
 20062       <newline> character. Similarly, since patterns are matched against individual lines of the input,  
 20063       there is no way for a pattern to match a <newline> character found in the input.

## 20064 OPTIONS

20065       The *grep* utility shall conform to the System Interface Definitions volume of  
 20066       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

20067       The following options shall be supported:

20068       **-E**       Match using extended regular expressions. Treat each pattern specified as an ERE,  
 20069       as described in the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 20070       Section 9.4, Extended Regular Expressions. If any entire ERE pattern matches some  
 20071       part of an input line, the line shall be matched. A null ERE shall match every line.

20072       **-F**       Match using fixed strings. Treat each pattern specified as a string instead of a  
 20073       regular expression. If an input line contains any of the patterns as a contiguous  
 20074       sequence of bytes, the line shall be matched. A null string shall match every line.

20075       **-c**       Write only a count of selected lines to standard output.20076       **-e *pattern\_list***

20077       Specify one or more patterns to be used during the search for input. The  
 20078       application shall ensure that patterns in *pattern\_list* are separated by a <newline>  
 20079       character. A null pattern can be specified by two adjacent <newline> characters in  
 20080       *pattern\_list*. Unless the **-E** or **-F** option is also specified, each pattern shall be  
 20081       treated as a BRE, as described in the System Interface Definitions volume of  
 20082       IEEE Std. 1003.1-200x, Section 9.3, Basic Regular Expressions. Multiple **-e** and **-f**  
 20083       options shall be accepted by the *grep* utility. All of the specified patterns shall be  
 20084       used when matching lines, but the order of evaluation is unspecified.

20085       **-f *pattern\_file***

20086       Read one or more patterns from the file named by the path name *pattern\_file*.  
 20087       Patterns in *pattern\_file* shall be terminated by a <newline> character. A null pattern

|       |                              |                                                                                                             |
|-------|------------------------------|-------------------------------------------------------------------------------------------------------------|
| 20088 |                              | can be specified by an empty line in <i>pattern_file</i> . Unless the <b>-E</b> or <b>-F</b> option is also |
| 20089 |                              | specified, each pattern shall be treated as a BRE, as described in the System                               |
| 20090 |                              | Interface Definitions volume of IEEE Std. 1003.1-200x, Section 9.3, Basic Regular                           |
| 20091 |                              | Expressions.                                                                                                |
| 20092 | <b>-i</b>                    | Perform pattern matching in searches without regard to case; see the System                                 |
| 20093 |                              | Interface Definitions volume of IEEE Std. 1003.1-200x, Section 9.2, Regular                                 |
| 20094 |                              | Expression General Requirements.                                                                            |
| 20095 | <b>-l</b>                    | (The letter ell.) Write only the names of files containing selected lines to standard                       |
| 20096 |                              | output. Path names shall be written once per file searched. If the standard input is                        |
| 20097 |                              | searched, a path name of "(standard input)" shall be written, in the POSIX                                  |
| 20098 |                              | locale. In other locales, "standard input" may be replaced by something more                                |
| 20099 |                              | appropriate in those locales.                                                                               |
| 20100 | <b>-n</b>                    | Precede each output line by its relative line number in the file, each file starting at                     |
| 20101 |                              | line 1. The line number counter shall be reset for each file processed.                                     |
| 20102 | <b>-q</b>                    | Quiet. Do not write anything to the standard output, regardless of matching lines.                          |
| 20103 |                              | Exit with zero status if an input line is selected.                                                         |
| 20104 | <b>-s</b>                    | Suppress the error messages ordinarily written for nonexistent or unreadable files.                         |
| 20105 |                              | Other error messages shall not be suppressed.                                                               |
| 20106 | <b>-v</b>                    | Select lines not matching any of the specified patterns. If the <b>-v</b> option is not                     |
| 20107 |                              | specified, selected lines shall be those that match any of the specified patterns.                          |
| 20108 | <b>-x</b>                    | Consider only input lines that use all characters in the line to match an entire fixed                      |
| 20109 |                              | string or regular expression to be matching lines.                                                          |
| 20110 | <b>OPERANDS</b>              |                                                                                                             |
| 20111 |                              | The following operands shall be supported:                                                                  |
| 20112 | <i>pattern</i>               | Specify one or more patterns to be used during the search for input. This operand                           |
| 20113 |                              | shall be treated as if it were specified as <b>-e pattern_list</b> .                                        |
| 20114 | <i>file</i>                  | A path name of a file to be searched for the patterns. If no <i>file</i> operands are                       |
| 20115 |                              | specified, the standard input shall be used.                                                                |
| 20116 | <b>STDIN</b>                 |                                                                                                             |
| 20117 |                              | The standard input shall be used only if no <i>file</i> operands are specified. See the INPUT FILES         |
| 20118 |                              | section.                                                                                                    |
| 20119 | <b>INPUT FILES</b>           |                                                                                                             |
| 20120 |                              | The input files shall be text files.                                                                        |
| 20121 | <b>ENVIRONMENT VARIABLES</b> |                                                                                                             |
| 20122 |                              | The following environment variables shall affect the execution of <i>grep</i> :                             |
| 20123 | <i>LANG</i>                  | Provide a default value for the internationalization variables that are unset or null.                      |
| 20124 |                              | If <i>LANG</i> is unset or null, the corresponding value from the implementation-                           |
| 20125 |                              | dependent default locale shall be used. If any of the internationalization variables                        |
| 20126 |                              | contains an invalid setting, the utility shall behave as if none of the variables had                       |
| 20127 |                              | been defined.                                                                                               |
| 20128 | <i>LC_ALL</i>                | If set to a non-empty string value, override the values of all the other                                    |
| 20129 |                              | internationalization variables.                                                                             |
| 20130 | <i>LC_COLLATE</i>            |                                                                                                             |
| 20131 |                              | Determine the locale for the behavior of ranges, equivalence classes and multi-                             |

20132 character collating elements within regular expressions.

20133 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 20134 characters (for example, single-byte as opposed to multi-byte characters in  
 20135 arguments and input files) and the behavior of character classes within regular  
 20136 expressions.

20137 **LC\_MESSAGES**  
 20138 Determine the locale that should be used to affect the format and contents of  
 20139 diagnostic messages written to standard error.

20140 XSI **NLSPATH** Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

20141 **ASYNCHRONOUS EVENTS**

20142 Default.

20143 **STDOUT**  
 20144 If the **-l** option is in effect, and the **-q** option is not, the following shall be written for each file  
 20145 containing at least one selected input line:  
 20146 "%s\n", <file>  
 20147 Otherwise, if more than one *file* argument appears, and **-q** is not specified, the *grep* utility shall  
 20148 prefix each output line by:  
 20149 "%s: ", <file>  
 20150 The remainder of each output line shall depend on the other options specified:  
 20151 • If the **-c** option is in effect, the remainder of each output line shall contain:  
 20152 "%d\n", <count>  
 20153 • Otherwise, if **-c** is not in effect and the **-n** option is in effect, the following shall be written to  
 20154 standard output:  
 20155 "%d: ", <line number>  
 20156 • Finally, the following shall be written to standard output:  
 20157 "%s", <selected-line contents>

20158 **STDERR**  
 20159 Used only for diagnostic messages.

20160 **OUTPUT FILES**  
 20161 None.

20162 **EXTENDED DESCRIPTION**  
 20163 None.

20164 **EXIT STATUS**  
 20165 The following exit values shall be returned:  
 20166 0 One or more lines were selected.  
 20167 1 No lines were selected.  
 20168 >1 An error occurred.

20169 **CONSEQUENCES OF ERRORS**

20170 If the **-q** option is specified, the exit status shall be zero if an input line is selected, even if an  
 20171 error was detected. Otherwise, default actions shall be performed.

20172 **APPLICATION USAGE**

20173 Care should be taken when using characters in *pattern\_list* that may also be meaningful to the  
 20174 command interpreter. It is safest to enclose the entire *pattern\_list* argument in single quotes:

20175 '...'

20176 The **-e** *pattern\_list* option has the same effect as the *pattern\_list* operand, but is useful when  
 20177 *pattern\_list* begins with the hyphen delimiter. It is also useful when it is more convenient to  
 20178 provide multiple patterns as separate arguments.

20179 Multiple **-e** and **-f** options are accepted and *grep* uses all of the patterns it is given while  
 20180 matching input text lines. (Note that the order of evaluation is not specified. If an  
 20181 implementation finds a null string as a pattern, it is allowed to use that pattern first, matching  
 20182 every line, and effectively ignore any other patterns.)

20183 The **-q** option provides a means of easily determining whether or not a pattern (or string) exists  
 20184 in a group of files. When searching several files, it provides a performance improvement  
 20185 (because it can quit as soon as it finds the first match) and requires less care by the user in  
 20186 choosing the set of files to supply as arguments (because it exits zero if it finds a match even if  
 20187 *grep* detected an access or read error on earlier file operands).

20188 **EXAMPLES**

20189 1. To find all uses of the word "Posix" (in any case) in file **text.mm** and write with line  
 20190 numbers:

20191 `grep -i -n posix text.mm`

20192 2. To find all empty lines in the standard input:

20193 `grep ^$`

20194 or:

20195 `grep -v .`

20196 3. Both of the following commands print all lines containing strings "abc" or "def" or both:

20197 `grep -E 'abc`  
 20198 `def'`

20199 `grep -F 'abc`  
 20200 `def'`

20201 4. Both of the following commands print all lines matching exactly "abc" or "def":

20202 `grep -E '^abc$`  
 20203 `^def$'`

20204 `grep -F -x 'abc`  
 20205 `def'`

20206 **RATIONALE**

20207 This *grep* has been enhanced in an upward-compatible way to provide the exact functionality of  
 20208 the historical *egrep* and *fgrep* commands as well. It was the clear intention of the standard  
 20209 developers to consolidate the three *greps* into a single command.

|       |                                                                                                                 |
|-------|-----------------------------------------------------------------------------------------------------------------|
| 20210 | The old <i>egrep</i> and <i>fgrep</i> commands are likely to be supported for many years to come as             |
| 20211 | implementation extensions, allowing historical applications to operate unmodified.                              |
| 20212 | Historical implementations usually silently ignored all but one of multiply-specified <b>-e</b> and <b>-f</b>   |
| 20213 | options, but were not consistent as to which specification was actually used.                                   |
| 20214 | The <b>-b</b> option was omitted from the <b>OPTIONS</b> section because block numbers are                      |
| 20215 | implementation-dependent.                                                                                       |
| 20216 | The System V restriction on using <b>-</b> to mean standard input was omitted.                                  |
| 20217 | A definition of action taken when given a null BRE or ERE is specified. This is an error condition              |
| 20218 | in some historical implementations.                                                                             |
| 20219 | The <b>-l</b> option previously indicated that its use was undefined when no files were explicitly              |
| 20220 | named. This behavior was historical and placed an unnecessary restriction on future                             |
| 20221 | implementations. It has been removed.                                                                           |
| 20222 | The historical BSD <i>grep</i> <b>-s</b> option practice is easily duplicated by redirecting standard output to |
| 20223 | <b>/dev/null</b> . The <b>-s</b> option required here is from System V.                                         |
| 20224 | The <b>-x</b> option, historically available only with <i>fgrep</i> , is available here for all of the non-     |
| 20225 | obsolescent versions.                                                                                           |
| 20226 | <b>FUTURE DIRECTIONS</b>                                                                                        |
| 20227 | None.                                                                                                           |
| 20228 | <b>SEE ALSO</b>                                                                                                 |
| 20229 | <i>sed</i>                                                                                                      |
| 20230 | <b>CHANGE HISTORY</b>                                                                                           |
| 20231 | First released in Issue 2.                                                                                      |
| 20232 | <b>Issue 4</b>                                                                                                  |
| 20233 | Aligned with the ISO/IEC 9945-2:1993 standard.                                                                  |
| 20234 | <b>Issue 6</b>                                                                                                  |
| 20235 | The Open Group corrigenda item U029/5 has been applied, correcting the SYNOPSIS.                                |
| 20236 | The normative text is reworded to avoid use of the term “must” for application requirements.                    |



20237 **NAME**

20238           hash — remember or report utility locations

20239 **SYNOPSIS**20240 xSI       hash [*utility...*]

20241           hash -r

20242

20243 **DESCRIPTION**

20244       The *hash* utility shall affect the way the current shell environment remembers the locations of  
 20245       utilities found as described in Section 2.9.1.1 on page 69. Depending on the arguments specified,  
 20246       it shall add utility locations to its list of remembered locations or it shall purge the contents of  
 20247       the list. When no arguments are specified, it shall report on the contents of the list.

20248       Utilities provided as built-ins to the shell shall not be reported by *hash*.20249 **OPTIONS**

20250       The *hash* utility shall conform to the System Interface Definitions volume of  
 20251       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

20252       The following option shall be supported:

20253       -r           Forget all previously remembered utility locations.

20254 **OPERANDS**

20255       The following operand shall be supported:

20256       *utility*       The name of a utility to be searched for and added to the list of remembered  
 20257       locations. If *utility* contains one or more slashes, the results are unspecified.

20258 **STDIN**

20259       Not used.

20260 **INPUT FILES**

20261       None.

20262 **ENVIRONMENT VARIABLES**20263       The following environment variables shall affect the execution of *hash*:

20264       *LANG*       Provide a default value for the internationalization variables that are unset or null.  
 20265       If *LANG* is unset or null, the corresponding value from the implementation-  
 20266       dependent default locale shall be used. If any of the internationalization variables  
 20267       contains an invalid setting, the utility shall behave as if none of the variables had  
 20268       been defined.

20269       *LC\_ALL*      If set to a non-empty string value, override the values of all the other  
 20270       internationalization variables.

20271       *LC\_CTYPE*   Determine the locale for the interpretation of sequences of bytes of text data as  
 20272       characters (for example, single-byte as opposed to multi-byte characters in  
 20273       arguments).

20274       *LC\_MESSAGES*

20275       Determine the locale that should be used to affect the format and contents of  
 20276       diagnostic messages written to standard error.

20277       *NLSPATH*     Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

20278       *PATH*        Determine the location of *utility*, as described in the System Interface Definitions  
 20279       volume of IEEE Std. 1003.1-200x, Chapter 8, Environment Variables.

20280 **ASYNCHRONOUS EVENTS**

20281 Default.

20282 **STDOUT**

20283 The standard output of *hash* shall be used when no arguments are specified. Its format is  
 20284 unspecified, but includes the path name of each utility in the list of remembered locations for the  
 20285 current shell environment. This list shall consist of those utilities named in previous *hash*  
 20286 invocations that have been invoked, and may contain those invoked and found through the  
 20287 normal command search process.

20288 **STDERR**

20289 Used only for diagnostic messages.

20290 **OUTPUT FILES**

20291 None.

20292 **EXTENDED DESCRIPTION**

20293 None.

20294 **EXIT STATUS**

20295 The following exit values shall be returned:

20296 0 Successful completion.

20297 &gt;0 An error occurred.

20298 **CONSEQUENCES OF ERRORS**

20299 Default.

20300 **APPLICATION USAGE**

20301 Since *hash* affects the current shell execution environment, it is always provided as a shell  
 20302 regular built-in. If it is called in a separate utility execution environment, such as one of the  
 20303 following:

```
20304 nohup hash -r
20305 find . -type f | xargs hash
```

20306 it does not affect the command search process of the caller's environment.

20307 The *hash* utility may be implemented as an alias—for example, *alias -t -*, in which case utilities  
 20308 found through normal command search are not listed by the *hash* command.

20309 The effects of *hash -r* can also be achieved portably by resetting the value of *PATH*; in the  
 20310 simplest form, this can be:

20311 `PATH="$PATH"`

20312 The use of *hash* with *utility* names is unnecessary for most applications, but may provide a  
 20313 performance improvement on a few implementations; normally, the hashing process is included  
 20314 by default.

20315 **EXAMPLES**

20316 None.

20317 **RATIONALE**

20318 None.

20319 **FUTURE DIRECTIONS**

20320 None.

20321 **SEE ALSO**

20322           Section 2.9.1.1 on page 69

20323 **CHANGE HISTORY**

20324           First released in Issue 2.

20325 **Issue 4**20326           Relocated from the *sh* description to reflect its status as a regular built-in utility.

20327 **NAME**

20328           *head* — copy the first part of files

20329 **SYNOPSIS**

20330           *head* [*-n number*][*file...*]

20331 **DESCRIPTION**

20332           The *head* utility shall copy its input files to the standard output, ending the output for each file at  
20333           a designated point.

20334           Copying shall end at the point in each input file indicated by the *-n number* option. The option-  
20335           argument *number* shall be counted in units of lines.

20336 **OPTIONS**

20337           The *head* utility shall conform to the System Interface Definitions volume of  
20338           IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

20339           The following option shall be supported:

20340           *-n number*   The first *number* lines of each input file shall be copied to standard output. The  
20341                           application shall ensure that the *number* option-argument is a positive decimal  
20342                           integer.

20343           If no options are specified, *head* shall act as if *-n 10* had been specified.

20344 **OPERANDS**

20345           The following operand shall be supported:

20346           *file*           A path name of an input file. If no *file* operands are specified, the standard input  
20347                           shall be used.

20348 **STDIN**

20349           The standard input shall be used only if no *file* operands are specified. See the INPUT FILES  
20350           section.

20351 **INPUT FILES**

20352           Input files shall be text files, but the line length is not restricted to {LINE\_MAX} bytes.

20353 **ENVIRONMENT VARIABLES**

20354           The following environment variables shall affect the execution of *head*:

20355           *LANG*           Provide a default value for the internationalization variables that are unset or null.  
20356                           If *LANG* is unset or null, the corresponding value from the implementation-  
20357                           dependent default locale shall be used. If any of the internationalization variables  
20358                           contains an invalid setting, the utility shall behave as if none of the variables had  
20359                           been defined.

20360           *LC\_ALL*        If set to a non-empty string value, override the values of all the other  
20361                           internationalization variables.

20362           *LC\_CTYPE*   Determine the locale for the interpretation of sequences of bytes of text data as  
20363                           characters (for example, single-byte as opposed to multi-byte characters in  
20364                           arguments and input files).

20365           *LC\_MESSAGES*

20366                           Determine the locale that should be used to affect the format and contents of  
20367                           diagnostic messages written to standard error.

20368 XSI       *NLSPATH*   Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

**20369 ASYNCHRONOUS EVENTS**

20370 Default.

**20371 STDOUT**

20372 The standard output shall contain designated portions of the input files.

20373 If multiple *file* operands are specified, *head* shall precede the output for each with the header:

20374 "\n==> %s <==\n", <pathname>

20375 except that the first header written shall not include the initial <newline> character.

**20376 STDERR**

20377 Used only for diagnostic messages.

**20378 OUTPUT FILES**

20379 None.

**20380 EXTENDED DESCRIPTION**

20381 None.

**20382 EXIT STATUS**

20383 The following exit values shall be returned:

20384 0 Successful completion.

20385 >0 An error occurred.

**20386 CONSEQUENCES OF ERRORS**

20387 Default.

**20388 APPLICATION USAGE**

20389 The obsolescent *–number* form is withdrawn in this version. Applications should use the *–n*  
20390 *number* option.

**20391 EXAMPLES**

20392 To write the first ten lines of all files (except those with a leading period) in the directory:

20393 head \*

**20394 RATIONALE**

20395 Although it is possible to simulate *head* with *sed 10q* for a single file, the standard developers  
20396 decided that the popularity of *head* on historical BSD systems warranted its inclusion alongside  
20397 *tail*.

20398 This standard version of *head* follows the Utility Syntax Guidelines. The *–n* option was added to  
20399 this new interface so that *head* and *tail* would be more logically related.

20400 There is no *–c* option (as there is in *tail*) because it is not historical practice and because other  
20401 utilities in this volume of IEEE Std. 1003.1-200x provide similar functionality.

**20402 FUTURE DIRECTIONS**

20403 None.

**20404 SEE ALSO**

20405 *sed*, *tail*

**20406 CHANGE HISTORY**

20407 First released in Issue 4.

|       |                                                                                              |  |
|-------|----------------------------------------------------------------------------------------------|--|
| 20408 | Issue 6                                                                                      |  |
| 20409 | The obsolescent – <b>number</b> form is withdrawn.                                           |  |
| 20410 | The normative text is reworded to avoid use of the term “must” for application requirements. |  |

20411 **NAME**

20412       iconv — codeset conversion

20413 **SYNOPSIS**20414       iconv [-cs] -f *fromcode* -t *tocode* [*file* ...]

20415       iconv -l

20416 **DESCRIPTION**

20417       The *iconv* utility shall convert the encoding of characters in *file* from one codeset to another and  
 20418       write the results to standard output.

20419       When the options indicate that charmap files are used to specify the codesets (see **OPTIONS**),  
 20420       the codeset conversion shall be accomplished by performing a logical join on the symbolic  
 20421       character names in the two charmaps. The implementation need not support the use of charmap  
 20422       files for codeset conversion unless the POSIX2\_LOCALEDEF symbol is defined on the system.

20423 **OPTIONS**

20424       The *iconv* utility shall conform to the System Interface Definitions volume of  
 20425       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

20426       The following options shall be supported:

20427       -c           Omit any invalid characters from the output. When -c is not used, the results of  
 20428       encountering invalid characters in the input stream (either those that are not valid  
 20429       members of the *fromcode* or those that have no corresponding value in *tocode*) shall  
 20430       be specified in the system documentation. The presence or absence of -c shall not  
 20431       affect the exit status of *iconv*.

20432       -f *fromcode* Identify the codeset of the input file. If the option-argument contains a slash  
 20433       character, *iconv* shall attempt to use it as the path name of a charmap file, as  
 20434       defined in the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 20435       Section 6.4, Character Set Description File. If the path name does not represent a  
 20436       valid, readable charmap file, the results are undefined. If the option-argument does  
 20437       not contain a slash, it shall be considered the name of one of the codeset  
 20438       descriptions provided by the system, in an unspecified format. The valid values of  
 20439       the option-argument without a slash are implementation-dependent. If this option  
 20440       is omitted, the codeset of the current locale shall be used.

20441       -l           Write all supported *fromcode* and *tocode* values to standard output in an unspecified  
 20442       format.

20443       -s           Suppress any messages written to standard error concerning invalid characters.  
 20444       When -s is not used, the results of encountering invalid characters in the input  
 20445       stream (either those that are not valid members of the *fromcode* or those that have  
 20446       no corresponding value in *tocode*) shall be specified in the system documentation.  
 20447       The presence or absence of -s shall not affect the exit status of *iconv*.

20448       -t *tocode*   Identify the codeset to be used for the output file. The semantics are equivalent to  
 20449       the -f *fromcode* option.

20450       If either -f or -t represents a charmap file, but the other does not (or is omitted), or both -f and  
 20451       -t are omitted, the results are undefined.

20452 **OPERANDS**

20453       The following operand shall be supported:

20454       *file*        A path name of an input file. If no *file* operands are specified, or if a *file* operand is  
 20455       '- ', the standard input shall be used.

20456 **STDIN**

20457       The standard input shall be used only if no *file* operands are specified, or if a file operand is '–'. |

20458 **INPUT FILES**

20459       The input file shall be a text file. |

20460 **ENVIRONMENT VARIABLES**

20461       The following environment variables shall affect the execution of *iconv*:

20462       *LANG*       Provide a default value for the internationalization variables that are unset or null.  
 20463                   If *LANG* is unset or null, the corresponding value from the implementation-  
 20464                   dependent default locale shall be used. If any of the internationalization variables  
 20465                   contains an invalid setting, the utility shall behave as if none of the variables had  
 20466                   been defined.

20467       *LC\_ALL*     If set to a non-empty string value, override the values of all the other  
 20468                   internationalization variables.

20469       *LC\_CTYPE*   Determine the locale for the interpretation of sequences of bytes of text data as  
 20470                   characters (for example, single-byte as opposed to multi-byte characters in  
 20471                   arguments). During translation of the file, this variable is superseded by the use of  
 20472                   the *fromcode* option-argument.

20473       *LC\_MESSAGES*

20474                   Determine the locale that should be used to affect the format and contents of  
 20475                   diagnostic messages written to standard error. |

20476 XSI       *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*. |

20477 **ASYNCHRONOUS EVENTS**

20478       Default.

20479 **STDOUT**

20480       When the *–l* option is used, the standard output shall contain all supported *fromcode* and *to*  
 20481       *code* values, written in an unspecified format. |

20482       When the *–l* option is not used, the standard output shall contain the sequence of characters  
 20483       read from the input files, translated to the specified codeset. Nothing else shall be written to the  
 20484       standard output. |

20485 **STDERR**

20486       Used only for diagnostic messages.

20487 **OUTPUT FILES**

20488       None.

20489 **EXTENDED DESCRIPTION**

20490       None.

20491 **EXIT STATUS**

20492       The following exit values shall be returned:

20493       0   Successful completion.

20494       >0   An error occurred.

20495 **CONSEQUENCES OF ERRORS**

20496       Default.



**20497 APPLICATION USAGE**

20498       The user must ensure that both charmap files use the same symbolic names for characters the  
20499       two codesets have in common.

**20500 EXAMPLES**

20501       The following example converts the contents of file **mail.x400** from the ISO/IEC 6937:1994  
20502       standard codeset to the ISO/IEC 8859-1:1998 standard codeset, and stores the results in file  
20503       **mail.local**:

20504       `iconv -f IS6937 -t IS8859 mail.x400 > mail.local`

**20505 RATIONALE**

20506       The *iconv* utility can be used portably only when the user provides two charmap files as option-  
20507       arguments. This is because a single charmap provided by the user cannot reliably be joined with  
20508       the names in a system-provided character set description. The valid values for *fromcode* and  
20509       *tocode* are implementation-dependent and do not have to have any relation to the charmap  
20510       mechanisms. As an aid to interactive users, the `-l` option was adopted from the Plan 9 operating  
20511       system. It writes information concerning these implementation-dependent values. The format is  
20512       unspecified because there are many possible useful formats that could be chosen, such as a  
20513       matrix of valid combinations of *fromcode* and *tocode*. The `-l` option is not intended for shell script  
20514       usage; portable applications will have to use charmaps.

**20515 FUTURE DIRECTIONS**

20516       None.

**20517 SEE ALSO**

20518       *gencat*

**20519 CHANGE HISTORY**

20520       First released in Issue 3.

**20521 Issue 4**

20522       Format reorganized.

20523       Utility Syntax Guidelines support mandated.

20524       Internationalized environment variable support mandated.

**20525 Issue 6**

20526       This utility has been rewritten to align with the IEEE P1003.2b draft standard. Specifically, the  
20527       ability to use charmap files for conversion has been added.

20528 **NAME**

20529       id — return user identity

20530 **SYNOPSIS**20531       id [*user*]20532       id -G[*-n*] [*user*]20533       id -g[*-nr*] [*user*]20534       id -u[*-nr*] [*user*]20535 **DESCRIPTION**

20536       If no *user* operand is provided, the *id* utility shall write the user and group IDs and the  
 20537       corresponding user and group names of the invoking process to standard output. If the effective  
 20538       and real IDs do not match, both shall be written. If multiple groups are supported by the  
 20539       underlying system (see the description of {NGROUPS\_MAX} in the System Interfaces volume of  
 20540       IEEE Std. 1003.1-200x), the supplementary group affiliations of the invoking process shall also be  
 20541       written.

20542       If a *user* operand is provided and the process has the appropriate privileges, the user and group  
 20543       IDs of the selected user shall be written. In this case, effective IDs shall be assumed to be  
 20544       identical to real IDs. If the selected user has more than one allowable group membership listed  
 20545       in the group database, these shall be written in the same manner as the supplementary groups  
 20546       described in the preceding paragraph.

20547 **OPTIONS**

20548       The *id* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 20549       Section 12.2, Utility Syntax Guidelines.

20550       The following options shall be supported:

20551       **-G**       Output all different group IDs (effective, real, and supplementary) only, using the  
 20552       format "%u\n". If there is more than one distinct group affiliation, output each  
 20553       such affiliation, using the format " %u", before the <newline> character is output.

20554       **-g**       Output only the effective group ID, using the format "%u\n".

20555       **-n**       Output the name in the format %s instead of the numeric ID using the format %u.

20556       **-r**       Output the real ID instead of the effective ID.

20557       **-u**       Output only the effective user ID, using the format "%u\n".

20558 **OPERANDS**

20559       The following operand shall be supported:

20560       *user*       The login name for which information is to be written.

20561 **STDIN**

20562       Not used.

20563 **INPUT FILES**

20564       None.

20565 **ENVIRONMENT VARIABLES**20566       The following environment variables shall affect the execution of *id*:

20567       **LANG**       Provide a default value for the internationalization variables that are unset or null.  
 20568       If *LANG* is unset or null, the corresponding value from the implementation-  
 20569       dependent default locale shall be used. If any of the internationalization variables  
 20570       contains an invalid setting, the utility shall behave as if none of the variables had

20571                   been defined.

20572           *LC\_ALL*     If set to a non-empty string value, override the values of all the other  
20573                   internationalization variables.

20574           *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
20575                   characters (for example, single-byte as opposed to multi-byte characters in  
20576                   arguments).

20577           *LC\_MESSAGES*  
20578                   Determine the locale that should be used to affect the format and contents of  
20579                   diagnostic messages written to standard error and informative messages written to  
20580                   standard output.

20581 XSI       *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

20582 **ASYNCHRONOUS EVENTS**

20583       Default.

20584 **STDOUT**

20585       The following formats shall be used when the *LC\_MESSAGES* locale category specifies the  
20586       POSIX locale. In other locales, the strings *uid*, *gid*, *eid*, *egid*, and *groups* may be replaced with  
20587       more appropriate strings corresponding to the locale.

20588       "uid=%u(%s) gid=%u(%s)\n", <real user ID>, <user-name>,  
20589        <real group ID>, <group-name>

20590       If the effective and real user IDs do not match, the following shall be inserted immediately  
20591       before the '\n' character in the previous format:

20592       " eid=%u(%s) "

20593       with the following arguments added at the end of the argument list:

20594       "effective user ID", <effective user-name>

20595       If the effective and real group IDs do not match, the following shall be inserted directly before  
20596       the '\n' character in the format string (and after any addition resulting from the effective and  
20597       real user IDs not matching):

20598       " egid=%u(%s) "

20599       with the following arguments added at the end of the argument list:

20600       <effective group-ID>, <effective group name>

20601       If the process has supplementary group affiliations or the selected user is allowed to belong to  
20602       multiple groups, the first shall be added directly before the <newline> character in the format  
20603       string:

20604       " groups=%u(%s) "

20605       with the following arguments added at the end of the argument list:

20606       <supplementary group ID>, <supplementary group name>

20607       and the necessary number of the following added after that for any remaining supplementary  
20608       group IDs:

20609       " ,%u(%s) "

20610       and the necessary number of the following arguments added at the end of the argument list:

20611 <supplementary group ID>, <supplementary group name>

20612 If any of the user ID, group ID, effective user ID, effective group ID, or supplementary/multiple  
20613 group IDs cannot be mapped by the system into printable user or group names, the  
20614 corresponding (%s) and name argument is omitted from the corresponding format string.

20615 When any of the options are specified, the output format shall be as described in the OPTIONS  
20616 section.

20617 **STDERR**

20618 Used only for diagnostic messages.

20619 **OUTPUT FILES**

20620 None.

20621 **EXTENDED DESCRIPTION**

20622 None.

20623 **EXIT STATUS**

20624 The following exit values shall be returned:

20625 0 Successful completion.

20626 >0 An error occurred.

20627 **CONSEQUENCES OF ERRORS**

20628 Default.

20629 **APPLICATION USAGE**

20630 Output produced by the **-G** option and by the default case could potentially produce very long  
20631 lines on systems that support large numbers of supplementary groups. (On systems with user  
20632 and group IDs that are 32-bit integers and with group names with a maximum of 8 bytes per  
20633 name, 93 supplementary groups plus distinct effective and real group and user IDs could  
20634 theoretically overflow the 2 048-byte {LINE\_MAX} text file line limit on the default output case.  
20635 It would take about 186 supplementary groups to overflow the 2 048-byte barrier using *id -G*).  
20636 This is not expected to be a problem in practice, but in cases where it is a concern, applications  
20637 should consider using *fold -s* before postprocessing the output of *id*.

20638 **EXAMPLES**

20639 None.

20640 **RATIONALE**

20641 The functionality provided by the 4 BSD *groups* utility can be simulated using:

20642 *id -Gn [ user ]*

20643 The 4 BSD command *groups* was considered, but it was not included because it did not provide  
20644 the functionality of the *id* utility of the SVID. Also, it was thought that it would be easier to  
20645 modify *id* to provide the additional functionality necessary to systems with multiple groups  
20646 than to invent another command.

20647 The options **-u**, **-g**, **-n**, and **-r** were added to ease the use of *id* with shell commands  
20648 substitution. Without these options it is necessary to use some preprocessor such as *sed* to select  
20649 the desired piece of information. Since output such as that produced by:

20650 *id -u -n*

20651 is frequently wanted, it seemed desirable to add the options.

20652 **FUTURE DIRECTIONS**

20653 None.

20654 **SEE ALSO**

20655 *fold*, *logname*, *who*, the System Interfaces volume of IEEE Std. 1003.1-200x, *getgid()*, *getgroups()*,  
20656 *getuid()*

20657 **CHANGE HISTORY**

20658 First released in Issue 2.

20659 **Issue 4**

20660 Aligned with the ISO/IEC 9945-2:1993 standard.

20661 **NAME**

20662       ipcrm — remove an XSI message queue, semaphore set, or shared memory segment identifier

20663 **SYNOPSIS**

```
20664 xsi ipcrm [-q msgid | -Q msgkey | -s semid | -S semkey |
20665 -m shmid | -M shmkey] ...
```

20666

20667 **DESCRIPTION**

20668       The *ipcrm* utility shall remove zero or more message queues, semaphore sets, or shared memory segments. The interprocess communication facilities to be removed are specified by the options.

20670       Only a user with appropriate privilege shall be allowed to remove an interprocess communication facility that was not created by or owned by the user invoking *ipcrm*.

20672 **OPTIONS**

20673       The *ipcrm* facility supports the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

20675       The following options shall be supported:

20676       **-q msgid**     Remove the message queue identifier *msgid* from the system and destroy the message queue and data structure associated with it.

20678       **-m shmid**     Remove the shared memory identifier *shmid* from the system. The shared memory segment and data structure associated with it shall be destroyed after the last detach.

20681       **-s semid**     Remove the semaphore identifier *semid* from the system and destroy the set of semaphores and data structure associated with it.

20683       **-Q msgkey**    Remove the message queue identifier, created with key *msgkey*, from the system and destroy the message queue and data structure associated with it.

20685       **-M shmkey**    Remove the shared memory identifier, created with key *shmkey*, from the system. The shared memory segment and data structure associated with it shall be destroyed after the last detach.

20688       **-S semkey**    Remove the semaphore identifier, created with key *semkey*, from the system and destroy the set of semaphores and data structure associated with it.

20690 **OPERANDS**

20691       None.

20692 **STDIN**

20693       Not used.

20694 **INPUT FILES**

20695       None.

20696 **ENVIRONMENT VARIABLES**

20697       The following environment variables shall affect the execution of *ipcrm*:

20698       **LANG**        Provide a default value for the internationalization variables that are unset or null. If *LANG* is unset or null, the corresponding value from the implementation-dependent default locale shall be used. If any of the internationalization variables contain an invalid setting, the utility behaves as if none of the variables had been set.

20703       **LC\_ALL**       If set to a non-empty string value, override the values of all the other internationalization variables.

20704

|       |                               |                                                                                                                                                                           |
|-------|-------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 20705 | <b>LC_CTYPE</b>               | Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments). |
| 20706 |                               |                                                                                                                                                                           |
| 20707 |                               |                                                                                                                                                                           |
| 20708 | <b>LC_MESSAGES</b>            |                                                                                                                                                                           |
| 20709 |                               | Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.                                              |
| 20710 |                               |                                                                                                                                                                           |
| 20711 | <b>NLSPATH</b>                |                                                                                                                                                                           |
| 20712 |                               | Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .                                                                                     |
| 20713 | <b>ASYNCHRONOUS EVENTS</b>    |                                                                                                                                                                           |
| 20714 |                               | Default.                                                                                                                                                                  |
| 20715 | <b>STDOUT</b>                 |                                                                                                                                                                           |
| 20716 |                               | Not used.                                                                                                                                                                 |
| 20717 | <b>STDERR</b>                 |                                                                                                                                                                           |
| 20718 |                               | Used only for diagnostic messages.                                                                                                                                        |
| 20719 | <b>OUTPUT FILES</b>           |                                                                                                                                                                           |
| 20720 |                               | None.                                                                                                                                                                     |
| 20721 | <b>EXTENDED DESCRIPTION</b>   |                                                                                                                                                                           |
| 20722 |                               | None.                                                                                                                                                                     |
| 20723 | <b>EXIT STATUS</b>            |                                                                                                                                                                           |
| 20724 |                               | The following exit values shall be returned:                                                                                                                              |
| 20725 | 0                             | Successful completion.                                                                                                                                                    |
| 20726 | >0                            | An error occurred.                                                                                                                                                        |
| 20727 | <b>CONSEQUENCES OF ERRORS</b> |                                                                                                                                                                           |
| 20728 |                               | Default.                                                                                                                                                                  |
| 20729 | <b>APPLICATION USAGE</b>      |                                                                                                                                                                           |
| 20730 |                               | None.                                                                                                                                                                     |
| 20731 | <b>EXAMPLES</b>               |                                                                                                                                                                           |
| 20732 |                               | None.                                                                                                                                                                     |
| 20733 | <b>RATIONALE</b>              |                                                                                                                                                                           |
| 20734 |                               | None.                                                                                                                                                                     |
| 20735 | <b>FUTURE DIRECTIONS</b>      |                                                                                                                                                                           |
| 20736 |                               | None.                                                                                                                                                                     |
| 20737 | <b>SEE ALSO</b>               |                                                                                                                                                                           |
| 20738 |                               | <i>ipcs</i> , the System Interfaces volume of IEEE Std. 1003.1-200x, <i>msgctl()</i> , <i>semctl()</i> , <i>shmctl()</i>                                                  |
| 20739 | <b>CHANGE HISTORY</b>         |                                                                                                                                                                           |
| 20740 |                               | First released in Issue 5.                                                                                                                                                |

20741 **NAME**

20742 *ipcs* — report XSI interprocess communication facilities status

20743 **SYNOPSIS**

20744 xsi *ipcs* [-qms] [-a | -bcopt]

20745

20746 **DESCRIPTION**

20747 The *ipcs* utility shall write information about active interprocess communication facilities.

20748 Without options, information shall be written in short format for message queues, shared  
20749 memory segments, and semaphores sets that are currently active in the system. Otherwise, the  
20750 information that is displayed is controlled by the options specified.

20751 **OPTIONS**

20752 The *ipcs* facility supports the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
20753 Section 12.2, Utility Syntax Guidelines.

20754 The *ipcs* utility accepts the following options:

20755 **-q** Write information about active message queues.

20756 **-m** Write information about active shared memory segments.

20757 **-s** Write information about active semaphores sets.

20758 If **-q**, **-m**, or **-s** are specified, only information about those facilities shall be written. If none of  
20759 these three are specified, information about all three shall be written subject to the following  
20760 options:

20761 **-a** Use all print options. (This is a shorthand notation for **-b**, **-c**, **-o**, **-p**, and **-t**.)

20762 **-b** Write information on maximum allowable size. (Maximum number of bytes in  
20763 messages on queue for message queues, size of segments for shared memory, and  
20764 number of semaphores in each set for semaphores.)

20765 **-c** Write creator's user name and group name; see below.

20766 **-o** Write information on outstanding usage. (Number of messages on queue and total  
20767 number of bytes in messages on queue for message queues, and number of  
20768 processes attached to shared memory segments.)

20769 **-p** Write process number information. (Process ID of last process to send a message  
20770 and process ID of last process to receive a message on message queues, process ID  
20771 of creating process, and process ID of last process to attach or detach on shared  
20772 memory segments.)

20773 **-t** Write time information. (Time of the last control operation that changed the access  
20774 permissions for all facilities, time of last *msgsnd()* and *msgrcv()* operations on  
20775 message queues, time of last *shmat()* and *shmdt()* operations on shared memory,  
20776 and time of last *semop()* operation on semaphores.)

20777 **OPERANDS**

20778 None.

20779 **STDIN**

20780 Not used.



20781 **INPUT FILES**

- 20782           • The group database
- 20783           • The user database

20784 **ENVIRONMENT VARIABLES**

20785           The following environment variables shall affect the execution of *ipcs*:

- 20786           *LANG*       Provide a default value for the internationalization variables that are unset or null.  
20787                       If *LANG* is unset or null, the corresponding value from the implementation-  
20788                       dependent default locale shall be used. If any of the internationalization variables  
20789                       contain an invalid setting, the utility behaves as if none of the variables had been  
20790                       set.
- 20791           *LC\_ALL*     If set to a non-empty string value, override the values of all the other  
20792                       internationalization variables.
- 20793           *LC\_CTYPE*   Determine the locale for the interpretation of sequences of bytes of text data as  
20794                       characters (for example, single-byte as opposed to multi-byte characters in  
20795                       arguments).
- 20796           *LC\_MESSAGES*  
20797                       Determine the locale that should be used to affect the format and contents of  
20798                       diagnostic messages written to standard error.
- 20799           *NLSPATH*   Determine the location of message catalogs for the processing of *LC\_MESSAGES*.
- 20800           *TZ*         Determine the timezone for the time strings written by *ipcs*.

20801 **ASYNCHRONOUS EVENTS**

20802           Default.

20803 **STDOUT**

20804           An introductory line shall be written with the format:

20805           "IPC status from %s as of %s\n", <source>, <date>

20806           where <source> indicates the source used to gather the statistics and <date> is the information  
20807           that would be produced by the *date* command when invoked in the POSIX locale.

20808           The *ipcs* utility then shall create up to three reports depending upon the *-q*, *-m*, and *-s* options.  
20809           The first report shall indicate the status of message queues, the second report shall indicate the  
20810           status of shared memory segments, and the third report shall indicate the status of semaphore  
20811           sets.

20812           If the corresponding facility is not installed or has not been used since the last reboot, then the  
20813           report shall be written out in the format:

20814           "%s facility not in system.\n", <facility>

20815           where <facility> is *Message Queue*, *Shared Memory*, or *Semaphore*, as appropriate. If the facility has  
20816           been installed and has been used since the last reboot, column headings separated by one or  
20817           more spaces and followed by a <newline> shall be written as indicated below followed by the  
20818           facility name written out using the format:

20819           "%s:\n", <facility>

20820           where <facility> is *Message Queues*, *Shared Memory*, or *Semaphores*, as appropriate. On the second  
20821           and third reports the column headings need not be written if the last column headings written  
20822           already provide column headings for all information in that report.

|       |                                                                                                   |              |                                                                                                     |
|-------|---------------------------------------------------------------------------------------------------|--------------|-----------------------------------------------------------------------------------------------------|
| 20823 | The column headings provided in the first column below and the meaning of the information in      |              |                                                                                                     |
| 20824 | those columns shall be given in order below; the letters in parentheses indicate the options that |              |                                                                                                     |
| 20825 | shall cause the corresponding column to appear; “all” means that the column shall always          |              |                                                                                                     |
| 20826 | appear. Each column is separated by one or more <space> characters. Note that these options       |              |                                                                                                     |
| 20827 | only determine what information is provided for each report; they do not determine which          |              |                                                                                                     |
| 20828 | reports are written.                                                                              |              |                                                                                                     |
| 20829 | T                                                                                                 | (all)        | Type of facility:                                                                                   |
| 20830 |                                                                                                   | q            | Message queue.                                                                                      |
| 20831 |                                                                                                   | m            | Shared memory segment.                                                                              |
| 20832 |                                                                                                   | s            | Semaphore.                                                                                          |
| 20833 |                                                                                                   |              | This field is a single character written using the format %c.                                       |
| 20834 | ID                                                                                                | (all)        | The identifier for the facility entry. This field shall be written using the format                 |
| 20835 |                                                                                                   |              | %d.                                                                                                 |
| 20836 | KEY                                                                                               | (all)        | The key used as an argument to <i>msgget()</i> , <i>semget()</i> , or <i>shmget()</i> to create the |
| 20837 |                                                                                                   |              | facility entry.                                                                                     |
| 20838 |                                                                                                   | <b>Note:</b> | The key of a shared memory segment is changed to IPC_PRIVATE                                        |
| 20839 |                                                                                                   |              | when the segment has been removed until all processes attached                                      |
| 20840 |                                                                                                   |              | to the segment detach it.                                                                           |
| 20841 |                                                                                                   |              | This field shall be written using the format 0x%x.                                                  |
| 20842 | MODE                                                                                              | (all)        | The facility access modes and flags. The mode shall consist of 11 characters                        |
| 20843 |                                                                                                   |              | that are interpreted as follows.                                                                    |
| 20844 |                                                                                                   |              | The first character shall be:                                                                       |
| 20845 |                                                                                                   | S            | If a process is waiting on a <i>msgsnd()</i> operation.                                             |
| 20846 |                                                                                                   | –            | If the above is not true.                                                                           |
| 20847 |                                                                                                   |              | The second character shall be:                                                                      |
| 20848 |                                                                                                   | R            | If a process is waiting on a <i>msgrcv()</i> operation.                                             |
| 20849 |                                                                                                   | C or –       | If the associated shared memory segment is to be cleared when the                                   |
| 20850 |                                                                                                   |              | first attach operation is executed.                                                                 |
| 20851 |                                                                                                   | –            | If none of the above is true.                                                                       |
| 20852 |                                                                                                   |              | The next nine characters shall be interpreted as three sets of three bits each.                     |
| 20853 |                                                                                                   |              | The first set refers to the owner's permissions; the next to permissions of                         |
| 20854 |                                                                                                   |              | others in the usergroup of the facility entry; and the last to all others. Within                   |
| 20855 |                                                                                                   |              | each set, the first character indicates permission to read, the second character                    |
| 20856 |                                                                                                   |              | indicates permission to write or alter the facility entry, and the last character is                |
| 20857 |                                                                                                   |              | a minus sign (‘–’).                                                                                 |
| 20858 |                                                                                                   |              | The permissions shall be indicated as follows:                                                      |
| 20859 |                                                                                                   | r            | If read permission is granted.                                                                      |
| 20860 |                                                                                                   | w            | If write permission is granted.                                                                     |
| 20861 |                                                                                                   | a            | If alter permission is granted.                                                                     |
| 20862 |                                                                                                   | –            | If the indicated permission is not granted.                                                         |

20863 The first character following the permissions specifies if there is an alternate  
 20864 or additional access control method associated with the facility. If there is no  
 20865 alternate or additional access control method associated with the facility, a  
 20866 single <space> character shall be written; otherwise, another printable  
 20867 character is written.

20868 OWNER (all) The user name of the owner of the facility entry. If the user name of the owner  
 20869 is found in the user database, at least the first eight column positions of the  
 20870 name shall be written using the format %s. Otherwise, the user ID of the  
 20871 owner shall be written using the format %d.

20872 GROUP (all) The group name of the owner of the facility entry. If the group name of the  
 20873 owner is found in the group database, at least the first eight column positions  
 20874 of the name shall be written using the format %s. Otherwise, the group ID of  
 20875 the owner shall be written using the format %d.

20876 The following nine columns shall be only written out for message queues:

20877 CREATOR (a,c) The user name of the creator of the facility entry. If the user name of the  
 20878 creator is found in the user database, at least the first eight column positions  
 20879 of the name shall be written using the format %s. Otherwise, the user ID of  
 20880 the creator shall be written using the format %d.

20881 CGROUP (a,c) The group name of the creator of the facility entry. If the group name of the  
 20882 creator is found in the group database, at least the first eight column positions  
 20883 of the name shall be written using the format %s. Otherwise, the group ID of  
 20884 the creator shall be written using the format %d.

20885 CBYTES (a,o) The number of bytes in messages currently outstanding on the associated  
 20886 message queue. This field shall be written using the format %d.

20887 QNUM (a,o) The number of messages currently outstanding on the associated message  
 20888 queue. This field shall be written using the format %d.

20889 QBYTES (a,b) The maximum number of bytes allowed in messages outstanding on the  
 20890 associated message queue. This field shall be written using the format %d.

20891 LSPID (a,p) The process ID of the last process to send a message to the associated queue.  
 20892 This field shall be written using the format:

20893 "%d", <pid>

20894 where <pid> is 0 if no message has been sent to the corresponding message  
 20895 queue; otherwise, <pid> shall be the process ID of the last process to send a  
 20896 message to the queue.

20897 LRPID (a,p) The process ID of the last process to receive a message from the associated  
 20898 queue. This field shall be written using the format:

20899 "%d", <pid>

20900 where <pid> is 0 if no message has been received from the corresponding  
 20901 message queue; otherwise, <pid> shall be the process ID of the last process to  
 20902 receive a message from the queue.

20903 STIME (a,t) The time the last message was sent to the associated queue. If a message has  
 20904 been sent to the corresponding message queue, the hour, minute, and second  
 20905 of the last time a message was sent to the queue shall be written using the  
 20906 format %d:%2.2d:%2.2d. Otherwise, the format " no-entry" shall be written.

|       |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
|-------|-----------------------------------------------------------------------------------|-------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 20907 | RTIME                                                                             | (a,t) | The time the last message was received from the associated queue. If a message has been received from the corresponding message queue, the hour, minute, and second of the last time a message was received from the queue shall be written using the format %d:%2.2d:%2.2d. Otherwise, the format " no-entry" shall be written. |
| 20908 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20909 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20910 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20911 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20912 | The following eight columns shall be only written out for shared memory segments. |       |                                                                                                                                                                                                                                                                                                                                  |
| 20913 | CREATOR                                                                           | (a,c) | The user of the creator of the facility entry. If the user name of the creator is found in the user database, at least the first eight column positions of the name shall be written using the format %s. Otherwise, the user ID of the creator shall be written using the format %d.                                            |
| 20914 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20915 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20916 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20917 | CGROUP                                                                            | (a,c) | The group name of the creator of the facility entry. If the group name of the creator is found in the group database, at least the first eight column positions of the name shall be written using the format %s. Otherwise, the group ID of the creator shall be written using the format %d.                                   |
| 20918 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20919 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20920 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20921 | NATTCH                                                                            | (a,o) | The number of processes attached to the associated shared memory segment. This field shall be written using the format %d.                                                                                                                                                                                                       |
| 20922 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20923 | SEGSZ                                                                             | (a,b) | The size of the associated shared memory segment. This field shall be written using the format %d.                                                                                                                                                                                                                               |
| 20924 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20925 | CPID                                                                              | (a,p) | The process ID of the creator of the shared memory entry. This field shall be written using the format %d.                                                                                                                                                                                                                       |
| 20926 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20927 | LPID                                                                              | (a,p) | The process ID of the last process to attach or detach the shared memory segment. This field shall be written using the format:                                                                                                                                                                                                  |
| 20928 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20929 |                                                                                   |       | "%d", <pid>                                                                                                                                                                                                                                                                                                                      |
| 20930 |                                                                                   |       | where <pid> is 0 if no process has attached the corresponding shared memory segment; otherwise, <pid> shall be the process ID of the last process to attach or detach the segment.                                                                                                                                               |
| 20931 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20932 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20933 | ATIME                                                                             | (a,t) | The time the last attach on the associated shared memory segment was completed. If the corresponding shared memory segment has ever been attached, the hour, minute, and second of the last time the segment was attached shall be written using the format %d:%2.2d:%2.2d. Otherwise, the format " no-entry" shall be written.  |
| 20934 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20935 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20936 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20937 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20938 | DTIME                                                                             | (a,t) | The time the last detach on the associated shared memory segment was completed. If the corresponding shared memory segment has ever been detached, the hour, minute, and second of the last time the segment was detached shall be written using the format %d:%2.2d:%2.2d. Otherwise, the format " no-entry" shall be written.  |
| 20939 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20940 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20941 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20942 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20943 | The following four columns shall be only written out for semaphore sets:          |       |                                                                                                                                                                                                                                                                                                                                  |
| 20944 | CREATOR                                                                           | (a,c) | The user of the creator of the facility entry. If the user name of the creator is found in the user database, at least the first eight column positions of the name shall be written using the format %s . Otherwise, the user ID of the creator shall be written using the format %d.                                           |
| 20945 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20946 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20947 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20948 | CGROUP                                                                            | (a,c) | The group name of the creator of the facility entry. If the group name of the creator is found in the group database, at least the first eight column positions of the name shall be written using the format %s. Otherwise, the group ID of the creator shall be written using the format %d.                                   |
| 20949 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20950 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |
| 20951 |                                                                                   |       |                                                                                                                                                                                                                                                                                                                                  |

20952 NSEMS (a,b) The number of semaphores in the set associated with the semaphore entry. |  
 20953 This field shall be written using the format %d.

20954 OTIME (a,t) The time the last semaphore operation on the set associated with the  
 20955 semaphore entry was completed. If a semaphore operation has ever been  
 20956 performed on the corresponding semaphore set, the hour, minute, and second  
 20957 of the last semaphore operation on the semaphore set shall be written using  
 20958 the format %d:%2.2d:%2.2d. Otherwise, the format " no-entry" shall be  
 20959 written.

20960 The following column shall be written for all three reports when it is requested:

20961 CTIME (a,t) The time the associated entry was created or changed. The hour, minute, and  
 20962 second of the time when the associated entry was created shall be written  
 20963 using the format %d:%2.2d:%2.2d.

20964 **STDERR**  
 20965 Used only for diagnostic messages.

20966 **OUTPUT FILES**  
 20967 None.

20968 **EXTENDED DESCRIPTION**  
 20969 None.

20970 **EXIT STATUS**  
 20971 The following exit values shall be returned:  
 20972 0 Successful completion.  
 20973 >0 An error occurred.

20974 **CONSEQUENCES OF ERRORS**  
 20975 Default.

20976 **APPLICATION USAGE**  
 20977 Things can change while *ipcs* is running; the information it gives is guaranteed to be accurate  
 20978 only when it was retrieved.

20979 **EXAMPLES**  
 20980 None.

20981 **RATIONALE**  
 20982 None.

20983 **FUTURE DIRECTIONS**  
 20984 None.

20985 **SEE ALSO**  
 20986 The System Interfaces volume of IEEE Std. 1003.1-200x, *msgop()*, *msgrcv()*, *msgsnd()*, *semget()*,  
 20987 *semop()*, *shmat()*, *shmdt()*, *shmget()*, *shmop()*

20988 **CHANGE HISTORY**  
 20989 First released in Issue 5.

20990 **Issue 6**  
 20991 The Open Group corrigenda item U020/1 has been applied, correcting the SYNOPSIS.  
 20992 The Open Group corrigenda items U032/1 and U032/2 have been applied, clarifying the output  
 20993 format. |

|             |       |                                                      |  |             |
|-------------|-------|------------------------------------------------------|--|-------------|
| <div></div> | 20994 | The Open Group Base Resolution bwg98-004 is applied. |  | <div></div> |
|-------------|-------|------------------------------------------------------|--|-------------|

20995 **NAME**20996       *jobs* — display status of jobs in the current session20997 **SYNOPSIS**20998 UP       *jobs* [-l | -p][*job\_id*...]  
2099921000 **DESCRIPTION**21001       The *jobs* utility shall display the status of jobs that were started in the current shell environment;  
21002       see Section 2.12 on page 90.21003       When *jobs* reports the termination status of a job, the shell shall remove its process ID from the  
21004       list of those “known in the current shell execution environment”; see Section 2.9.3.1 on page 74.21005 **OPTIONS**21006       The *jobs* utility shall conform to the System Interface Definitions volume of  
21007       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

21008       The following options shall be supported:

21009       -l       (The letter ell.) Provide more information about each job listed. This information  
21010       shall include the job number, current job, process group ID, state, and the  
21011       command that formed the job.

21012       -p       Display only the process IDs for the process group leaders of the selected jobs.

21013       By default, the *jobs* utility shall display the status of all stopped jobs, running background jobs  
21014       and all jobs whose status has changed and have not been reported by the shell.21015 **OPERANDS**

21016       The following operand shall be supported:

21017       *job\_id*       Specifies the jobs for which the status is to be displayed. If no *job\_id* is given, the  
21018       status information for all jobs shall be displayed. The format of *job\_id* is described  
21019       in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 3.207,  
21020       Job Control Job ID.21021 **STDIN**

21022       Not used.

21023 **INPUT FILES**

21024       None.

21025 **ENVIRONMENT VARIABLES**21026       The following environment variables shall affect the execution of *jobs*:21027       LANG       Provide a default value for the internationalization variables that are unset or null.  
21028       If *LANG* is unset or null, the corresponding value from the implementation-  
21029       dependent default locale shall be used. If any of the internationalization variables  
21030       contains an invalid setting, the utility shall behave as if none of the variables had  
21031       been defined.21032       LC\_ALL       If set to a non-empty string value, override the values of all the other  
21033       internationalization variables.21034       LC\_CTYPE     Determine the locale for the interpretation of sequences of bytes of text data as  
21035       characters (for example, single-byte as opposed to multi-byte characters in  
21036       arguments).

21037       LC\_MESSAGES

21038       Determine the locale that should be used to affect the format and contents of

21039 diagnostic messages written to standard error and informative messages written to  
 21040 standard output.

21041 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

## 21042 ASYNCHRONOUS EVENTS

21043 Default.

## 21044 STDOUT

21045 If the **-p** option is specified, the output shall consist of one line for each process ID:

21046 "%d\n", <process ID>

21047 Otherwise, if the **-l** option is not specified, the output shall be a series of lines of the form:

21048 "[%d] %c %s %s\n", <job-number>, <current>, <state>, <command>

21049 where the fields shall be as follows:

21050 <current> The character '+' identifies the job that would be used as a default for the *fg* or *bg*  
 21051 utilities; this job can also be specified using the *job\_id*%+ or "%%". The character  
 21052 '-' identifies the job that would become the default if the current default job were  
 21053 to exit; this job can also be specified using the *job\_id*%-. For other jobs, this field is  
 21054 a <space> character. At most one job can be identified with '+' and at most one  
 21055 job can be identified with '-'. If there is any suspended job, then the current job  
 21056 shall be a suspended job. If there are at least two suspended jobs, then the previous  
 21057 job also shall be a suspended job.

21058 <job-number> A number that can be used to identify the process group to the *wait*, *fg*, *bg*, and *kill*  
 21059 utilities. Using these utilities, the job can be identified by prefixing the job number  
 21060 with '% '.

21061 <state> One of the following strings (in the POSIX locale):

21062 **Running** Indicates that the job has not been suspended by a signal and has not  
 21063 exited.

21064 **Done** Indicates that the job completed and returned exit status zero.

21065 **Done(code)** Indicates that the job completed normally and that it exited with the  
 21066 specified non-zero exit status, *code*, expressed as a decimal number.

21067 **Stopped** Indicates that the job was suspended by the SIGTSTP signal.

21068 **Stopped (SIGTSTP)**

21069 Indicates that the job was suspended by the SIGTSTP signal.

21070 **Stopped (SIGSTOP)**

21071 Indicates that the job was suspended by the SIGSTOP signal.

21072 **Stopped (SIGTTIN)**

21073 Indicates that the job was suspended by the SIGTTIN signal.

21074 **Stopped (SIGTTOU)**

21075 Indicates that the job was suspended by the SIGTTOU signal.

21076 The implementation may substitute the string **Suspended** in place of **Stopped**. If  
 21077 the job was terminated by a signal, the format of <state> is unspecified, but it shall  
 21078 be visibly distinct from all of the other <state> formats shown here and shall  
 21079 indicate the name or description of the signal causing the termination.



21080           <*command*>   The associated command that was given to the shell.

21081           If the **-l** option is specified, a field containing the process group ID shall be inserted before the

21082           <*state*> field. Also, more processes in a process group may be output on separate lines, using

21083           only the process ID and <*command*> fields.

21084 **STDERR**

21085           Used only for diagnostic messages.

21086 **OUTPUT FILES**

21087           None.

21088 **EXTENDED DESCRIPTION**

21089           None.

21090 **EXIT STATUS**

21091           The following exit values shall be returned:

21092           0   Successful completion.

21093           >0   An error occurred.

21094 **CONSEQUENCES OF ERRORS**

21095           Default.

21096 **APPLICATION USAGE**

21097           The **-p** option is the only portable way to find out the process group of a job because different

21098           implementations have different strategies for defining the process group of the job. Usage such

21099           as *\$jobs -p* provides a way of referring to the process group of the job in an implementation-

21100           independent way.

21101           The *jobs* utility does not work as expected when it is operating in its own utility execution

21102           environment because that environment has no applicable jobs to manipulate. See the

21103           APPLICATION USAGE section for *bg* on page 243. For this reason, *jobs* is generally

21104           implemented as a shell regular built-in.

21105           Application writers should note that this utility need not be provided on systems that do not

21106           support the User Portability Utilities option.

21107 **EXAMPLES**

21108           None.

21109 **RATIONALE**

21110           Both "%%" and "%+" are used to refer to the current job. Both forms are of equal validity—the

21111           "%%" mirroring "\$\$" and "%+" mirroring the output of *jobs*. Both forms reflect historical

21112           practice of the KornShell and the C shell with job control.

21113           The extensions to the shell specified in this volume of IEEE Std. 1003.1-200x have mostly been

21114           based on features provided by the KornShell. The job control features provided by *bg*, *fg*, and *jobs*

21115           are also based on the KornShell. The standard developers examined the characteristics of the C

21116           shell versions of these utilities and found that differences exist. Despite widespread use of the C

21117           shell, the KornShell versions were selected for this volume of IEEE Std. 1003.1-200x to maintain a

21118           degree of uniformity with the rest of the KornShell features selected (such as the very popular

21119           command line editing features).

21120           The *jobs* utility is not dependent on the job control option, as are the seemingly related *bg* and *fg*

21121           utilities because *jobs* is useful for examining background jobs, regardless of the condition of job

21122           control. When the user has invoked a *set +m* command and job control has been turned off, *jobs*

21123           can still be used to examine the background jobs associated with that current session. Similarly,

21124           *kill* can then be used to kill background jobs with *kill% <background job number>*.

- 21125 The output for terminated jobs is left unspecified to accommodate various historical systems.  
21126 The following formats have been witnessed:
- 21127 1. **Killed**(*signal name*)
  - 21128 2. *signal name*
  - 21129 3. *signal name*(**coredump**)
  - 21130 4. *signal description*– **core dumped**
- 21131 Most users should be able to understand these formats, although it means that applications have  
21132 trouble parsing them.
- 21133 The calculation of job IDs was not described since this would suggest an implementation, which  
21134 may impose unnecessary restrictions.
- 21135 In an early proposal, a **-n** option was included to “Display the status of jobs that have changed,  
21136 exited, or stopped since the last status report”. It was removed because the shell always writes  
21137 any changed status of jobs before each prompt.
- 21138 **FUTURE DIRECTIONS**
- 21139 None.
- 21140 **SEE ALSO**
- 21141 *bg, fg, kill, wait*
- 21142 **CHANGE HISTORY**
- 21143 First released in Issue 4.
- 21144 **Issue 6**
- 21145 This utility is now marked as part of the User Portability Utilities option.
- 21146 The JC shading is removed as job control is mandatory in this issue.

21147 **NAME**

21148           join — relational database operator

21149 **SYNOPSIS**

21150           join [-a *file\_number* | -v *file\_number*][-e *string*][-o *list*][-t *char*]  
 21151               [-1 *field*][-2 *field*] *file1 file2*

21152 **DESCRIPTION**

21153           The *join* utility shall perform an equality join on the files *file1* and *file2*. The joined files shall be  
 21154           written to the standard output.

21155           The join field is a field in each file on which the files are compared. By default, *join* shall write  
 21156           one line in the output for each pair of lines in *file1* and *file2* that have identical join fields. The  
 21157           output line by default shall consist of the join field, then the remaining fields from *file1*, then the  
 21158           remaining fields from *file2*. This format can be changed by using the -o option (see below). The  
 21159           -a option can be used to add unmatched lines to the output. The -v option can be used to output  
 21160           only unmatched lines.

21161 **Notes to Reviewers**21162           *This section with side shading will not appear in the final copy. - Ed.*

21163           D1, XCU, ERN 265 proposes to add the following text here: "If the same key appears more than  
 21164           once in either file, all possible pairwise combinations are output, in unspecified order".

21165           By default, the files *file1* and *file2* should be ordered in the collating sequence of *sort -b* on the  
 21166           fields on which they shall be joined, by default the first in each line. All selected output shall be  
 21167           written in the same collating sequence.

21168           The default input field separators shall be <blank> characters. In this case, multiple separators  
 21169           shall count as one field separator, and leading separators shall be ignored. The default output  
 21170           field separator shall be a <space> character.

21171           The field separator and collating sequence can be changed by using the -t option (see below).

21172           If the input files are not in the appropriate collating sequence, the results are unspecified.

21173 **OPTIONS**

21174           The *join* utility shall conform to the System Interface Definitions volume of  
 21175           IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

21176           The following options shall be supported:

21177           -a *file\_number*

21178               Produce a line for each unpairable line in file *file\_number*, where *file\_number* is 1 or  
 21179               2, in addition to the default output. If both -a1 and -a2 are specified, all unpairable  
 21180               lines shall be output.

21181           -e *string*       Replace empty output fields in the list selected by -o with the string *string*.

21182           -o *list*           Construct the output line to comprise the fields specified in *list*, each element of  
 21183           which shall have one of the following two forms:

- 21184               1. *file\_number.field*, where *file\_number* is a file number and *field* is a decimal  
 21185               integer field number
- 21186               2. 0 (zero), representing the join field

21187           The elements of *list* shall be either comma-separated or <blank>-separated, as  
 21188           specified in Guideline 8 of the System Interface Definitions volume of  
 21189           IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines. The fields specified

|       |                              |                                                                                                               |  |
|-------|------------------------------|---------------------------------------------------------------------------------------------------------------|--|
| 21190 |                              | by <i>list</i> shall be written for all selected output lines. Fields selected by <i>list</i> that do         |  |
| 21191 |                              | not appear in the input shall be treated as empty output fields. (See the <b>-e</b> option.)                  |  |
| 21192 |                              | Only specifically requested fields shall be written. The application shall ensure that                        |  |
| 21193 |                              | <i>list</i> is a single command line argument.                                                                |  |
| 21194 | <b>-t char</b>               | Use character <i>char</i> as a separator, for both input and output. Every appearance of                      |  |
| 21195 |                              | <i>char</i> in a line shall be significant. When this option is specified, the collating                      |  |
| 21196 |                              | sequence should be the same as <i>sort</i> without the <b>-b</b> option.                                      |  |
| 21197 | <b>-v file_number</b>        |                                                                                                               |  |
| 21198 |                              | Instead of the default output, produce a line only for each unpairable line in                                |  |
| 21199 |                              | <i>file_number</i> , where <i>file_number</i> is 1 or 2. If both <b>-v1</b> and <b>-v2</b> are specified, all |  |
| 21200 |                              | unpairable lines shall be output.                                                                             |  |
| 21201 | <b>-1 field</b>              | Join on the <i>field</i> th field of file 1. Fields are decimal integers starting with 1.                     |  |
| 21202 | <b>-2 field</b>              | Join on the <i>field</i> th field of file 2. Fields are decimal integers starting with 1.                     |  |
| 21203 | <b>OPERANDS</b>              |                                                                                                               |  |
| 21204 |                              | The following operands shall be supported:                                                                    |  |
| 21205 | <i>file1, file2</i>          |                                                                                                               |  |
| 21206 |                              | A path name of a file to be joined. If either of the <i>file1</i> or <i>file2</i> operands is '-', the        |  |
| 21207 |                              | standard input shall be used in its place.                                                                    |  |
| 21208 | <b>STDIN</b>                 |                                                                                                               |  |
| 21209 |                              | The standard input shall be used only if the <i>file1</i> or <i>file2</i> operand is '-'. See the INPUT FILES |  |
| 21210 |                              | section.                                                                                                      |  |
| 21211 | <b>INPUT FILES</b>           |                                                                                                               |  |
| 21212 |                              | The input files shall be text files.                                                                          |  |
| 21213 | <b>ENVIRONMENT VARIABLES</b> |                                                                                                               |  |
| 21214 |                              | The following environment variables shall affect the execution of <i>join</i> :                               |  |
| 21215 | <b>LANG</b>                  | Provide a default value for the internationalization variables that are unset or null.                        |  |
| 21216 |                              | If <b>LANG</b> is unset or null, the corresponding value from the implementation-                             |  |
| 21217 |                              | dependent default locale shall be used. If any of the internationalization variables                          |  |
| 21218 |                              | contains an invalid setting, the utility shall behave as if none of the variables had                         |  |
| 21219 |                              | been defined.                                                                                                 |  |
| 21220 | <b>LC_ALL</b>                | If set to a non-empty string value, override the values of all the other                                      |  |
| 21221 |                              | internationalization variables.                                                                               |  |
| 21222 | <b>LC_COLLATE</b>            |                                                                                                               |  |
| 21223 |                              | Determine the locale of the collating sequence <i>join</i> expects to have been used when                     |  |
| 21224 |                              | the input files were sorted.                                                                                  |  |
| 21225 | <b>LC_CTYPE</b>              | Determine the locale for the interpretation of sequences of bytes of text data as                             |  |
| 21226 |                              | characters (for example, single-byte as opposed to multi-byte characters in                                   |  |
| 21227 |                              | arguments and input files).                                                                                   |  |
| 21228 | <b>LC_MESSAGES</b>           |                                                                                                               |  |
| 21229 |                              | Determine the locale that should be used to affect the format and contents of                                 |  |
| 21230 |                              | diagnostic messages written to standard error.                                                                |  |
| 21231 | <b>XLSPATH</b>               | Determine the location of message catalogs for the processing of <b>LC_MESSAGES</b> .                         |  |

21232 **ASYNCHRONOUS EVENTS**

21233 Default.

21234 **STDOUT**

21235 The *join* utility output shall be a concatenation of selected character fields. When the **-o** option  
 21236 is not specified, the output shall be:

21237 "%s%s%s\n", <join field>, <other file1 fields>,  
 21238 <other file2 fields>

21239 If the join field is not the first field in a file, the <other file fields> for that file shall be:

21240 <fields preceding join field>, <fields following join field>

21241 When the **-o** option is specified, the output format shall be:

21242 "%s\n", <concatenation of fields>

21243 where the concatenation of fields is described by the **-o** option, above.

21244 For either format, each field (except the last) shall be written with its trailing separator character.  
 21245 If the separator is the default (<blank> characters), a single <space> character shall be written  
 21246 after each field (except the last).

21247 **STDERR**

21248 Used only for diagnostic messages.

21249 **OUTPUT FILES**

21250 None.

21251 **EXTENDED DESCRIPTION**

21252 None.

21253 **EXIT STATUS**

21254 The following exit values shall be returned:

21255 0 All input files were output successfully.

21256 >0 An error occurred.

21257 **CONSEQUENCES OF ERRORS**

21258 Default.

21259 **APPLICATION USAGE**

21260 Path names consisting of numeric digits or of the form *string.string* should not be specified  
 21261 directly following the **-o** list.

21262 **EXAMPLES**

21263 The **-o 0** field essentially selects the union of the join fields. For example, given file **phone**:

|       |         |                 |
|-------|---------|-----------------|
| 21264 | !Name   | Phone Number    |
| 21265 | Don     | +1 123-456-7890 |
| 21266 | Hal     | +1 234-567-8901 |
| 21267 | Yasushi | +2 345-678-9012 |

21268 and file **fax**:

|       |         |                 |
|-------|---------|-----------------|
| 21269 | !Name   | Fax Number      |
| 21270 | Don     | +1 123-456-7899 |
| 21271 | Keith   | +1 456-789-0122 |
| 21272 | Yasushi | +2 345-678-9011 |

21273 (where the large expanses of white space are meant to each represent a single <tab> character),  
 21274 the command:

21275 `join -t "<tab>" -a 1 -a 2 -e '(unknown)' -o 0,1.2,2.2 phone fax`

21276 would produce:

| 21277 | !Name   | Phone Number    | Fax Number      |
|-------|---------|-----------------|-----------------|
| 21278 | Don     | +1 123-456-7890 | +1 123-456-7899 |
| 21279 | Hal     | +1 234-567-8901 | (unknown)       |
| 21280 | Keith   | (unknown)       | +1 456-789-0122 |
| 21281 | Yasushi | +2 345-678-9012 | +2 345-678-9011 |

## 21282 **Notes to Reviewers**

21283 *This section with side shading will not appear in the final copy. - Ed.*

21284 D1, XCU, ERN 265 proposes to add the following example.

21285 The following:

```

21286 fa:
21287 a x
21288 a y
21289 a z
21290 fb:
21291 a p
21292 a q

```

21293 would produce:

```

21294 a x p
21295 a x q
21296 a y p
21297 a y q
21298 a z p
21299 a z q

```

## 21300 **RATIONALE**

21301 The standard developers believed that *join* should operate as documented in the SVID and BSD  
 21302 not as historically implemented. Most implementations do not parse the **-o** option as described  
 21303 in their own documentation, and parse the elements as separate *argv* items until the item is not  
 21304 of the form *file\_number.field*. Early proposals indicated that undefined behavior would result if  
 21305 numeric file names were used immediately following the obsolescent multiple-argument form  
 21306 of the **-o** list. However, since *join* always requires two file name arguments, there never should  
 21307 be any ambiguity about whether an argument is associated with **-o** or not, and this application  
 21308 restriction is no longer present.

21309 The **-e** option is only effective when used with **-o** because, unless specific fields are identified  
 21310 using **-o**, *join* is not aware of what fields might be empty. The exception to this is the join field,  
 21311 but identifying an empty join field with the **-e** string is not historical practice and some scripts  
 21312 might break if this were changed.

21313 The 0 field in the **-o** list was adopted from the Tenth Edition version of *join* to satisfy  
 21314 international objections that the *join* in the base documents do not support the “full join” or  
 21315 “outer join” described in relational database literature. Although it has been possible to include  
 21316 a join field in the output (by default, or by field number using **-o**), the join field could not be  
 21317 included for an unpaired line selected by **-a**. The **-o 0** field essentially selects the union of the

- 21318 join fields.
- 21319 This sort of outer join was not possible with the *join* commands in the base documents. The **-o 0**  
 21320 field was chosen because it is an upward-compatible change for applications. An alternative was  
 21321 considered: have the join field represent the union of the fields in the files (where they are  
 21322 identical for matched lines, and one or both are null for unmatched lines). This was not adopted  
 21323 because it would break some historical applications.
- 21324 The obsolescent **-j**, **-j1**, and **-j2** options have been removed in this draft. Early proposals  
 21325 showed **-jfile\_number** field, but a space was never allowed before the *file\_number* and two  
 21326 option-arguments were never intended.
- 21327 The ability to specify *file2* as **-** is not historical practice; it was added for completeness.
- 21328 The **-v** option is not historical practice, but was considered necessary because it permitted the  
 21329 writing of *only* those lines that do not match on the join field, as opposed to the **-a** option, which  
 21330 prints both lines that do and do not match. This additional facility is parallel with the **-v** option  
 21331 of *grep*.
- 21332 Some historical implementations have been encountered where a blank line in one of the input  
 21333 files was considered to be the end of the file; the description in this volume of  
 21334 IEEE Std. 1003.1-200x does not cite this as an allowable case.
- 21335 **FUTURE DIRECTIONS**
- 21336 None.
- 21337 **SEE ALSO**
- 21338 *awk, comm, sort, uniq*
- 21339 **CHANGE HISTORY**
- 21340 First released in Issue 2.
- 21341 **Issue 4**
- 21342 Aligned with the ISO/IEC 9945-2: 1993 standard.
- 21343 **Issue 6**
- 21344 The obsolescent **-j** options and the multi-argument **-o** option are withdrawn in this issue.
- 21345 The normative text is reworded to avoid use of the term “must” for application requirements.

## 21346 NAME

21347 kill — terminate or signal processes

## 21348 SYNOPSIS

21349 kill -s *signal\_name* *pid*...21350 kill -l [*exit\_status*]

## 21351 DESCRIPTION

21352 The *kill* utility shall send a signal to the process or processes specified by each *pid* operand.

21353 For each *pid* operand, the *kill* utility shall perform actions equivalent to the *kill()* function  
 21354 defined in the System Interfaces volume of IEEE Std. 1003.1-200x called with the following  
 21355 arguments:

- 21356 • The value of the *pid* operand shall be used as the *pid* argument.
- 21357 • The *sig* argument is the value specified by the -s option, -*signal\_number* option, or the  
 21358 -*signal\_name* option, or by SIGTERM, if none of these options is specified.

## 21359 OPTIONS

21360 The *kill* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 21361 Section 12.2, Utility Syntax Guidelines.

21362 The following options shall be supported:

21363 -l (The letter ell.) Write all values of *signal\_name* supported by the implementation, if  
 21364 no operand is given. If an *exit\_status* operand is given and it is a value of the ' ? '  
 21365 shell special parameter (see Section 2.5.2 on page 43 and *wait* on page 1084)  
 21366 corresponding to a process that was terminated by a signal, the *signal\_name*  
 21367 corresponding to the signal that terminated the process shall be written. If an  
 21368 *exit\_status* operand is given and it is the unsigned decimal integer value of a signal  
 21369 number, the *signal\_name* (the symbolic constant name without the **SIG** prefix  
 21370 defined in the System Interface Definitions volume of IEEE Std. 1003.1-200x)  
 21371 corresponding to that signal shall be written. Otherwise, the results are  
 21372 unspecified.

21373 -s *signal\_name*

21374 Specify the signal to send, using one of the symbolic names defined in the  
 21375 <**signal.h**> header defined in the System Interface Definitions volume of  
 21376 IEEE Std. 1003.1-200x, Chapter 13, Headers. Values of *signal\_name* shall be  
 21377 recognized in a case-independent fashion, without the **SIG** prefix. In addition, the  
 21378 symbolic name 0 shall be recognized, representing the signal value zero. The  
 21379 corresponding signal shall be sent instead of SIGTERM.

## 21380 OPERANDS

21381 The following operands shall be supported:

21382 *pid* One of the following:

- 21383 1. A decimal integer specifying a process or process group to be signaled. The  
 21384 process or processes selected by positive, negative and zero values of the *pid*  
 21385 operand shall be as described for the *kill()* function defined in the System  
 21386 Interfaces volume of IEEE Std. 1003.1-200x. If process number 0 is specified,  
 21387 all processes in the current process group are signaled. For the effects of  
 21388 negative *pid* numbers, see the *kill()* function defined in the System Interfaces  
 21389 volume of IEEE Std. 1003.1-200x. If the first *pid* operand is negative, it should  
 21390 be preceded by "—" to keep it from being interpreted as an option.



|       |                              |                                                                                                                                            |  |
|-------|------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|--|
| 21391 |                              | 2. A job control job ID (see the System Interface Definitions volume of                                                                    |  |
| 21392 |                              | IEEE Std. 1003.1-200x, Section 3.207, Job Control Job ID) that identifies a                                                                |  |
| 21393 |                              | background process group to be signaled. The job control job ID notation is                                                                |  |
| 21394 |                              | applicable only for invocations of <i>kill</i> in the current shell execution                                                              |  |
| 21395 |                              | environment; see Section 2.12 on page 90.                                                                                                  |  |
| 21396 | <i>exit_status</i>           | A decimal integer specifying a signal number or the exit status of a process                                                               |  |
| 21397 |                              | terminated by a signal.                                                                                                                    |  |
| 21398 | <b>STDIN</b>                 |                                                                                                                                            |  |
| 21399 |                              | Not used.                                                                                                                                  |  |
| 21400 | <b>INPUT FILES</b>           |                                                                                                                                            |  |
| 21401 |                              | None.                                                                                                                                      |  |
| 21402 | <b>ENVIRONMENT VARIABLES</b> |                                                                                                                                            |  |
| 21403 |                              | The following environment variables shall affect the execution of <i>kill</i> :                                                            |  |
| 21404 | <i>LANG</i>                  | Provide a default value for the internationalization variables that are unset or null.                                                     |  |
| 21405 |                              | If <i>LANG</i> is unset or null, the corresponding value from the implementation-                                                          |  |
| 21406 |                              | dependent default locale shall be used. If any of the internationalization variables                                                       |  |
| 21407 |                              | contains an invalid setting, the utility shall behave as if none of the variables had                                                      |  |
| 21408 |                              | been defined.                                                                                                                              |  |
| 21409 | <i>LC_ALL</i>                | If set to a non-empty string value, override the values of all the other                                                                   |  |
| 21410 |                              | internationalization variables.                                                                                                            |  |
| 21411 | <i>LC_CTYPE</i>              | Determine the locale for the interpretation of sequences of bytes of text data as                                                          |  |
| 21412 |                              | characters (for example, single-byte as opposed to multi-byte characters in                                                                |  |
| 21413 |                              | arguments).                                                                                                                                |  |
| 21414 | <i>LC_MESSAGES</i>           |                                                                                                                                            |  |
| 21415 |                              | Determine the locale that should be used to affect the format and contents of                                                              |  |
| 21416 |                              | diagnostic messages written to standard error.                                                                                             |  |
| 21417 | XSI <i>NLSPATH</i>           | Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .                                                      |  |
| 21418 | <b>ASYNCHRONOUS EVENTS</b>   |                                                                                                                                            |  |
| 21419 |                              | Default.                                                                                                                                   |  |
| 21420 | <b>STDOUT</b>                |                                                                                                                                            |  |
| 21421 |                              | When the <i>-l</i> option is not specified, the standard output shall not be used.                                                         |  |
| 21422 |                              | When the <i>-l</i> option is specified, the symbolic name of each signal shall be written in the                                           |  |
| 21423 |                              | following format:                                                                                                                          |  |
| 21424 |                              | <code>"%s%c", &lt;signal_name&gt;, &lt;separator&gt;</code>                                                                                |  |
| 21425 |                              | where the <i>&lt;signal_name&gt;</i> is in uppercase, without the <b>SIG</b> prefix, and the <i>&lt;separator&gt;</i> shall be             |  |
| 21426 |                              | either a <i>&lt;newline&gt;</i> character or a <i>&lt;space&gt;</i> character. For the last signal written, <i>&lt;separator&gt;</i> shall |  |
| 21427 |                              | be a <i>&lt;newline&gt;</i> character.                                                                                                     |  |
| 21428 |                              | When both the <i>-l</i> option and <i>exit_status</i> operand are specified, the symbolic name of the                                      |  |
| 21429 |                              | corresponding signal shall be written in the following format:                                                                             |  |
| 21430 |                              | <code>"%s\n", &lt;signal_name&gt;</code>                                                                                                   |  |

21431 **STDERR**

21432       Used only for diagnostic messages.

21433 **OUTPUT FILES**

21434       None.

21435 **EXTENDED DESCRIPTION**

21436       None.

21437 **EXIT STATUS**

21438       The following exit values shall be returned:

21439       0   At least one matching process was found for each *pid* operand, and the specified signal was  
 21440       successfully processed for at least one matching process.

21441       &gt;0   An error occurred.

21442 **CONSEQUENCES OF ERRORS**

21443       Default.

21444 **APPLICATION USAGE**21445       Process numbers can be found by using *ps*.

21446       The job control job ID notation is not required to work as expected when *kill* is operating in its  
 21447       own utility execution environment. In either of the following examples:

```
21448 nohup kill %1 &
21449 system("kill %1");
```

21450       the *kill* operates in a different environment and does not share the shell's understanding of job  
 21451       numbers.

21452 **EXAMPLES**

21453       Any of the commands:

```
21454 kill -s kill 100 -165
21455 kill -s KILL 100 -165
```

21456       sends the SIGKILL signal to the process whose process ID is 100 and to all processes whose  
 21457       process group ID is 165, assuming the sending process has permission to send that signal to the  
 21458       specified processes, and that they exist.

21459       The System Interfaces volume of IEEE Std. 1003.1-200x and this volume of IEEE Std. 1003.1-200x  
 21460       do not require specific signal numbers for any *signal\_names*. Even the *-signal\_number* option  
 21461       provides symbolic (although numeric) names for signals. If a process is terminated by a signal,  
 21462       its exit status indicates the signal that killed it, but the exact values are not specified. The *kill -l*  
 21463       option, however, can be used to map decimal signal numbers and exit status values into the  
 21464       name of a signal. The following example reports the status of a terminated job:

```
21465 job
21466 stat=$?
21467 if [$stat -eq 0]
21468 then
21469 echo job completed successfully.
21470 elif [$stat -gt 128]
21471 then
21472 echo job terminated by signal SIG$(kill -l $stat).
21473 else
21474 echo job terminated with error code $stat.
21475 fi
```

21476 To avoid an ambiguity of an initial negative number argument specifying either a signal number  
 21477 or a process group, the ISO/IEC 9945-2: 1993 standard mandates that it always be considered the  
 21478 former. Therefore, to send the default signal to a process group (say 123), an application should  
 21479 use a command similar to one of the following:

21480 `kill -TERM -123`

21481 `kill -- -123`

#### 21482 **RATIONALE**

21483 The **-I** option originated from the C shell, and is also implemented in the KornShell. The C shell  
 21484 output can consist of multiple output lines because the signal names do not always fit on a  
 21485 single line on some terminal screens. The KornShell output also included the implementation-  
 21486 dependent signal numbers and was considered by the standard developers to be too difficult for  
 21487 scripts to parse conveniently. The specified output format is intended not only to accommodate  
 21488 the historical C shell output, but also to permit an entirely vertical or entirely horizontal listing  
 21489 on systems for which this is appropriate.

21490 An early proposal invented the name *SIGNULL* as a *signal\_name* for signal 0 (used by the System  
 21491 Interfaces volume of IEEE Std. 1003.1-200x to test for the existence of a process without sending  
 21492 it a signal). Since the *signal\_name* 0 can be used in this case unambiguously, *SIGNULL* has been  
 21493 removed.

21494 An early proposal also required symbolic *signal\_names* to be recognized with or without the **SIG**  
 21495 prefix. Historical versions of *kill* have not written the **SIG** prefix for the **-I** option and have not  
 21496 recognized the **SIG** prefix on *signal\_names*. Since neither application portability nor ease-of-use  
 21497 would be improved by requiring this extension, it is no longer required.

21498 This volume of IEEE Std. 1003.1-200x contains no utility that browses for process IDs. Values for  
 21499 *pid* are available via the '!' and '\$' parameters of the shell command language.

21500 The **-s** option was added in response to international interest in providing some form of *kill* that  
 21501 meets the Utility Syntax Guidelines.

21502 The job control job ID notation is not required to work as expected when *kill* is operating in its  
 21503 own utility execution environment. In either of the following examples:

21504 `nohup kill %1 &`  
 21505 `system("kill %1");`

21506 the *kill* operates in a different environment and does not understand how the shell has managed  
 21507 its job numbers.

#### 21508 **FUTURE DIRECTIONS**

21509 None.

#### 21510 **SEE ALSO**

21511 *ps*, *wait*, the System Interfaces volume of IEEE Std. 1003.1-200x, *kill()*, *<signal.h>*

#### 21512 **CHANGE HISTORY**

21513 First released in Issue 2.

#### 21514 **Issue 4**

21515 Aligned with the ISO/IEC 9945-2: 1993 standard.

#### 21516 **Issue 6**

21517 The obsolescent versions of the SYNOPSIS are withdrawn in this issue.

21518 **NAME**21519       lex — generate programs for lexical tasks (**DEVELOPMENT**)21520 **SYNOPSIS**21521 CD       lex -c [-t][-n] -v [*file* ...]

21522

21523 **DESCRIPTION**

21524       The *lex* utility shall generate C programs to be used in lexical processing of character input, and  
 21525       that can be used as an interface to *yacc*. The C programs shall be generated from *lex* source code  
 21526       and conform to the ISO C standard. Usually, the *lex* utility shall write the program it generates to  
 21527       the file **lex.yy.c**; the state of this file is unspecified if *lex* exits with a non-zero exit status. See the  
 21528       EXTENDED DESCRIPTION section for a complete description of the *lex* input language.

21529 **OPTIONS**

21530       The *lex* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 21531       Section 12.2, Utility Syntax Guidelines.

21532       The following options shall be supported:

21533       **-n**       Suppress the summary of statistics usually written with the **-v** option. If no table  
 21534       sizes are specified in the *lex* source code and the **-v** option is not specified, then **-n**  
 21535       is implied.

21536       **-t**       Write the resulting program to standard output instead of **lex.yy.c**.

21537       **-v**       Write a summary of *lex* statistics to the standard output. (See the discussion of *lex*  
 21538       table sizes in **Definitions in lex** on page 570.) If the **-t** option is specified and **-n** is  
 21539       not specified, this report shall be written to standard error. If table sizes are  
 21540       specified in the *lex* source code, and if the **-n** option is not specified, the **-v** option  
 21541       may be enabled.

21542 **OPERANDS**

21543       The following operand shall be supported:

21544       *file*       A path name of an input file. If more than one such *file* is specified, all files shall be  
 21545       concatenated to produce a single *lex* program. If no *file* operands are specified, or if  
 21546       a *file* operand is **'-'**, the standard input shall be used.

21547 **STDIN**

21548       The standard input shall be used if no *file* operands are specified, or if a *file* operand is **'-'**. See  
 21549       INPUT FILES.

21550 **INPUT FILES**

21551       The input files shall be text files containing *lex* source code, as described in the EXTENDED  
 21552       DESCRIPTION section.

21553 **ENVIRONMENT VARIABLES**

21554       If this variable is not set to the POSIX locale, the results are unspecified.

21555       The following environment variables shall affect the execution of *lex*:

21556       **LANG**       Provide a default value for the internationalization variables that are unset or null.  
 21557       If **LANG** is unset or null, the corresponding value from the implementation-  
 21558       dependent default locale shall be used. If any of the internationalization variables  
 21559       contains an invalid setting, the utility shall behave as if none of the variables had  
 21560       been defined.

21561       **LC\_ALL**      If set to a non-empty string value, override the values of all the other  
 21562       internationalization variables.

- 21563 **LC\_COLLATE**  
 21564 Determine the locale for the behavior of ranges, equivalence classes and multi-  
 21565 character collating elements within regular expressions. If this variable is not set to  
 21566 the POSIX locale, the results are unspecified.
- 21567 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 21568 characters (for example, single-byte as opposed to multi-byte characters in  
 21569 arguments and input files), and the behavior of character classes within regular  
 21570 expressions. If this variable is not set to the POSIX locale, the results are  
 21571 unspecified.
- 21572 **LC\_MESSAGES**  
 21573 Determine the locale that should be used to affect the format and contents of  
 21574 diagnostic messages written to standard error.
- 21575 XSI **NLSPATH** Determine the location of message catalogs for the processing of **LC\_MESSAGES**.
- 21576 **ASYNCHRONOUS EVENTS**  
 21577 Default.
- 21578 **STDOUT**  
 21579 If the **-t** option is specified, the text file of C source code output of *lex* shall be written to  
 21580 standard output.
- 21581 If the **-t** option is not specified:
- 21582 • Implementation-dependent informational, error, and warning messages concerning the  
 21583 contents of *lex* source code input shall be written to either the standard output or standard  
 21584 error.
  - 21585 • If the **-v** option is specified and the **-n** option is not specified, *lex* statistics shall also be  
 21586 written to either the standard output or standard error, in an implementation-dependent  
 21587 format. These statistics may also be generated if table sizes are specified with a **'%'** operator  
 21588 in the *Definitions* section, as long as the **-n** option is not specified.
- 21589 **STDERR**  
 21590 If the **-t** option is specified, implementation-dependent informational, error, and warning  
 21591 messages concerning the contents of *lex* source code input shall be written to the standard error.
- 21592 If the **-t** option is not specified:
- 21593 1. Implementation-dependent informational, error, and warning messages concerning the  
 21594 contents of *lex* source code input shall be written to either the standard output or standard  
 21595 error.
  - 21596 2. If the **-v** option is specified and the **-n** option is not specified, *lex* statistics shall also be  
 21597 written to either the standard output or standard error, in an implementation-dependent  
 21598 format. These statistics may also be generated if table sizes are specified with a **'%'**  
 21599 operator in the *Definitions* section, as long as the **-n** option is not specified.
- 21600 **OUTPUT FILES**  
 21601 A text file containing C source code shall be written to **lex.yy.c**, or to the standard output if the  
 21602 **-t** option is present.
- 21603 **EXTENDED DESCRIPTION**  
 21604 Each input file contains *lex* source code, which is a table of regular expressions with  
 21605 corresponding actions in the form of C program fragments.
- 21606 When **lex.yy.c** is compiled and linked with the *lex* library (using the **-ll** operand with *c89* or *cc*),  
 21607 the resulting program reads character input from the standard input and partitions it into strings

21608 that match the given expressions.

21609 When an expression is matched, these actions shall occur:

- 21610 • The input string that was matched is left in *yytext* as a null-terminated string; *yytext* is either
- 21611 an external character array or a pointer to a character string. As explained in **Definitions in**
- 21612 **lex**, the type can be explicitly selected using the **%array** or **%pointer** declarations, but the
- 21613 default is implementation-dependent.
- 21614 • The external **int** *yylen* is set to the length of the matching string.
- 21615 • The expression's corresponding program fragment, or action, is executed.

21616 During pattern matching, *lex* shall search the set of patterns for the single longest possible

21617 match. Among rules that match the same number of characters, the rule given first shall be

21618 chosen.

21619 The general format of *lex* source shall be:

```

21620 Definitions
21621 %%
21622 Rules
21623 %%
21624 UserSubroutines
```

21625 The first "%%" is required to mark the beginning of the rules (regular expressions and actions);

21626 the second "%%" is required only if user subroutines follow.

21627 Any line in the *Definitions* section beginning with a <blank> character shall be assumed to be a C

21628 program fragment and shall be copied to the external definition area of the **lex.yy.c** file.

21629 Similarly, anything in the *Definitions* section included between delimiter lines containing only

21630 "%{" and "%}" shall also be copied unchanged to the external definition area of the **lex.yy.c** file.

21631 Any such input (beginning with a <blank> character or within "%{" and "%}" delimiter lines)

21632 appearing at the beginning of the *Rules* section before any rules are specified shall be written to

21633 **lex.yy.c** after the declarations of variables for the *yylex* function and before the first line of code

21634 in *yylex*. Thus, user variables local to *yylex* can be declared here, as well as application code to

21635 execute upon entry to *yylex*.

21636 The action taken by *lex* when encountering any input beginning with a <blank> character or

21637 within "%{" and "%}" delimiter lines appearing in the *Rules* section but coming after one or

21638 more rules is undefined. The presence of such input may result in an erroneous definition of the

21639 *yylex* function.

## 21640 **Definitions in lex**

21641 *Definitions* appear before the first "%%" delimiter. Any line in this section not contained between

21642 "%{" and "%}" lines and not beginning with a <blank> character shall be assumed to define a

21643 *lex* substitution string. The format of these lines shall be:

```

21644 name substitute
```

21645 If a *name* does not meet the requirements for identifiers in the ISO C standard, the result is

21646 undefined. The string *substitute* shall replace the string {*name*} when it is used in a rule. The *name*

21647 string shall be recognized in this context only when the braces are provided and when it does

21648 not appear within a bracket expression or within double-quotes.

21649 In the *Definitions* section, any line beginning with a '%' (percent sign) character and followed by

21650 an alphanumeric word beginning with either 's' or 'S' shall define a set of start conditions.

21651 Any line beginning with a '%' followed by a word beginning with either 'x' or 'X' shall define

a set of exclusive start conditions. When the generated scanner is in a "%s" state, patterns with no state specified shall be also active; in a "%x" state, such patterns shall not be active. The rest of the line, after the first word, shall be considered to be one or more <blank> character-separated names of start conditions. Start condition names shall be constructed in the same way as definition names. Start conditions can be used to restrict the matching of regular expressions to one or more states as described in **Regular Expressions in lex** on page 572.

Implementations shall accept either of the following two mutually exclusive declarations in the *Definitions* section:

**%array**      Declare the type of *yytext* to be a null-terminated character array.

**%pointer**    Declare the type of *yytext* to be a pointer to a null-terminated character string.

The default type of *yytext* is implementation-dependent. If an application refers to *yytext* outside of the scanner source file (that is, via an **extern**), the application shall include the appropriate **%array** or **%pointer** declaration in the scanner source file.

Implementations shall accept declarations in the *Definitions* section for setting certain internal table sizes. The declarations are shown in the following table.

**Table 4-9** Table Size Declarations in *lex*

| Declaration | Description                        | Minimum Value |
|-------------|------------------------------------|---------------|
| <b>%p n</b> | Number of positions                | 2 500         |
| <b>%n n</b> | Number of states                   | 500           |
| <b>%a n</b> | Number of transitions              | 2 000         |
| <b>%e n</b> | Number of parse tree nodes         | 1 000         |
| <b>%k n</b> | Number of packed character classes | 1 000         |
| <b>%o n</b> | Size of the output array           | 3 000         |

In the table, *n* represents a positive decimal integer, preceded by one or more <blank> characters. The exact meaning of these table size numbers is implementation-dependent. The implementation shall document how these numbers affect the *lex* utility and how they are related to any output that may be generated by the implementation should space limitations be encountered during the execution of *lex*. It shall be possible to determine from this output which of the table size values needs to be modified to permit *lex* to successfully generate tables for the input language. The values in the column Minimum Value represent the lowest values conforming implementations shall provide.

### Rules in *lex*

The rules in *lex* source files are a table in which the left column contains regular expressions and the right column contains actions (C program fragments) to be executed when the expressions are recognized.

*ERE action*

*ERE action*

...

The extended regular expression (*ERE*) portion of a row shall be separated from *action* by one or more <blank> characters. A regular expression containing <blank> characters shall be recognized under one of the following conditions:

- The entire expression appears within double-quotes.
- The <blank> characters appear within double-quotes or square brackets.

- Each <blank> character is preceded by a backslash character.

## 21696 User Subroutines in lex

21697 Anything in the user subroutines section shall be copied to **lex.yy.c** following *yylex*.

## 21698 Regular Expressions in lex

21699 The *lex* utility shall support the set of extended regular expressions (see the System Interface  
21700 Definitions volume of IEEE Std. 1003.1-200x, Section 9.4, Extended Regular Expressions), with  
21701 the following additions and exceptions to the syntax:

21702 " . . . " Any string enclosed in double-quotes shall represent the characters within the  
21703 double-quotes as themselves, except that backslash escapes (which appear in the  
21704 following table) shall be recognized. Any backslash-escape sequence shall be  
21705 terminated by the closing quote. For example, "\01" "1" represents a single  
21706 string: the octal value 1 followed by the character '1'.

21707 <state>*r*, <state1, state2, . . .>*r*

21708 The regular expression *r* shall be matched only when the program is in one of the  
21709 start conditions indicated by *state*, *state1*, and so on; see **Actions in lex** on page 574.  
21710 (As an exception to the typographical conventions of the rest of this volume of  
21711 IEEE Std. 1003.1-200x, in this case <state> does not represent a metavariable, but  
21712 the literal angle-bracket characters surrounding a symbol.) The start condition  
21713 shall be recognized as such only at the beginning of a regular expression.

21714 *r/x* The regular expression *r* shall be matched only if it is followed by an occurrence of  
21715 regular expression *x* (*x* is the instance of trailing context, further defined below).  
21716 The token returned in *yytext* shall only match *r*. If the trailing portion of *r* matches  
21717 the beginning of *x*, the result is unspecified. The *r* expression cannot include  
21718 further trailing context or the '\$' (match-end-of-line) operator; *x* cannot include  
21719 the '^' (match-beginning-of-line) operator, nor trailing context, nor the '\$'  
21720 operator. That is, only one occurrence of trailing context is allowed in a *lex* regular  
21721 expression, and the '^' operator only can be used at the beginning of such an  
21722 expression.

21723 {*name*} When *name* is one of the substitution symbols from the *Definitions* section, the  
21724 string, including the enclosing braces, shall be replaced by the *substitute* value. The  
21725 *substitute* value shall be treated in the extended regular expression as if it were  
21726 enclosed in parentheses. No substitution shall occur if {*name*} occurs within a  
21727 bracket expression or within double-quotes.

21728 Within an ERE, a backslash character shall be considered to begin an escape sequence as  
21729 specified in the table in the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
21730 Chapter 5, File Format Notation ('\\', '\a', '\b', '\f', '\n', '\r', '\t', '\v'). In  
21731 addition, the escape sequences in the following table shall be recognized.

21732 A literal <newline> character cannot occur within an ERE; the escape sequence '\n' can be  
21733 used to represent a <newline> character. A <newline> character shall not be matched by a  
21734 period operator.



Table 4-10 Escape Sequences in *lex*

| Escape Sequence       | Description                                                                                                                                                                                                                                                                                                                                                        | Meaning                                                                                                                                                                                                                                                                                                                                                                             |
|-----------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>\digits</code>  | A backslash character followed by the longest sequence of one, two, or three octal-digit characters (01234567). If all of the digits are 0, (that is, representation of the NUL character), the behavior is undefined.                                                                                                                                             | The character whose encoding is represented by the one, two, or three-digit octal integer. If the size of a byte on the system is greater than nine bits, the valid escape sequence used to represent a byte is implementation-dependent. Multi-byte characters require multiple, concatenated escape sequences of this type, including the leading <code>'\'</code> for each byte. |
| <code>\xdigits</code> | A backslash character followed by the longest sequence of hexadecimal-digit characters (01234567abcdefABCDEF). If all of the digits are 0, (that is, representation of the NUL character), the behavior is undefined.                                                                                                                                              | The character whose encoding is represented by the hexadecimal integer.                                                                                                                                                                                                                                                                                                             |
| <code>\c</code>       | A backslash character followed by any character not described in this table or in the table in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 5, File Format Notation ( <code>'\'</code> , <code>'\a'</code> , <code>'\b'</code> , <code>'\f'</code> , <code>'\n'</code> , <code>'\r'</code> , <code>'\t'</code> , <code>'\v'</code> ). | The character <code>'c'</code> , unchanged.                                                                                                                                                                                                                                                                                                                                         |

The order of precedence given to extended regular expressions for *lex* differs from that specified in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 9.4, Extended Regular Expressions. The order of precedence for *lex* shall be as shown in the following table, from high to low.

**Note:** The escaped characters entry is not meant to imply that these are operators, but they are included in the table to show their relationships to the true operators. The start condition, trailing context, and anchoring notations have been omitted from the table because of the placement restrictions described in this section; they can only appear at the beginning or ending of an ERE.

Table 4-11 ERE Precedence in *lex*

| Extended Regular Expression              | Precedence            |
|------------------------------------------|-----------------------|
| <i>collation-related bracket symbols</i> | [ = ] [ : : ] [ . . ] |
| <i>escaped characters</i>                | \<special character>  |
| <i>bracket expression</i>                | [ ]                   |
| <i>quoting</i>                           | " . . . "             |
| <i>grouping</i>                          | ( )                   |
| <i>definition</i>                        | { name }              |
| <i>single-character RE duplication</i>   | * + ?                 |
| <i>concatenation</i>                     |                       |
| <i>interval expression</i>               | { m , n }             |
| <i>alternation</i>                       |                       |

The ERE anchoring operators '*^*' and '*\$*' do not appear in the table. With *lex* regular expressions, these operators are restricted in their use: the '*^*' operator can only be used at the beginning of an entire regular expression, and the '*\$*' operator only at the end. The operators apply to the entire regular expression. Thus, for example, the pattern "*(^abc)|(def\$)*" is undefined; it can instead be written as two separate rules, one with the regular expression "*^abc*" and one with "*def\$*", which share a common action via the special '*|*' action (see below). If the pattern were written "*^abc|def\$*", it would match either "*abc*" or "*def*" on a line by itself.

Unlike the general ERE rules, embedded anchoring is not allowed by most historical *lex* implementations. An example of embedded anchoring would be for patterns such as "*(^| )foo( |\$)*" to match "*foo*" when it exists as a complete word. This functionality can be obtained using existing *lex* features:

```

^foo/[\n] |
" foo"/[\n] /* Found foo as a separate word. */

```

Note also that '*\$*' is a form of trailing context (it is equivalent to "*/\n*") and as such cannot be used with regular expressions containing another instance of the operator (see the preceding discussion of trailing context).

The additional regular expressions trailing-context operator '*/'*' can be used as an ordinary character if presented within double-quotes, "*/"*"; preceded by a backslash, "*\"*"; or within a bracket expression, "*[ / ]*". The start-condition '*<*' and '*>*' operators shall be special only in a start condition at the beginning of a regular expression; elsewhere in the regular expression they shall be treated as ordinary characters.

### Actions in *lex*

The action to be taken when an ERE is matched can be a C program fragment or the special actions described below; the program fragment can contain one or more C statements, and can also include special actions. The empty C statement '*;*' shall be a valid action; any string in the *lex.yy.c* input that matches the pattern portion of such a rule is effectively ignored or skipped. However, the absence of an action shall not be valid, and the action *lex* takes in such a condition is undefined.

The specification for an action, including C statements and special actions, can extend across several lines if enclosed in braces:

```

ERE <one or more blanks> { program statement
 program statement }

```

21821 The default action when a string in the input to a **lex.yy.c** program is not matched by any  
 21822 expression shall be to copy the string to the output. Because the default behavior of a program  
 21823 generated by *lex* is to read the input and copy it to the output, a minimal *lex* source program that  
 21824 has just "%%" shall generate a C program that simply copies the input to the output unchanged.

21825 Four special actions shall be available:

21826 | ECHO; REJECT; BEGIN

21827 | The action ' | ' means that the action for the next rule is the action for this rule.  
 21828 Unlike the other three actions, ' | ' cannot be enclosed in braces or be semicolon-  
 21829 terminated; the application shall ensure that it is specified alone, with no other  
 21830 actions.

21831 **ECHO;** Write the contents of the string *yytext* on the output.

21832 **REJECT;** Usually only a single expression is matched by a given string in the input. **REJECT**  
 21833 means "continue to the next expression that matches the current input", and shall  
 21834 cause whatever rule was the second choice after the current rule to be executed for  
 21835 the same input. Thus, multiple rules can be matched and executed for one input  
 21836 string or overlapping input strings. For example, given the regular expressions  
 21837 "xyz" and "xy" and the input "xyz", usually only the regular expression "xyz"  
 21838 would match. The next attempted match would start after **z**. If the last action in the  
 21839 "xyz" rule is **REJECT**, both this rule and the "xy" rule would be executed. The  
 21840 **REJECT** action may be implemented in such a fashion that flow of control does not  
 21841 continue after it, as if it were equivalent to a **goto** to another part of *yylex*. The use  
 21842 of **REJECT** may result in somewhat larger and slower scanners.

21843 **BEGIN** The action:

21844 BEGIN *newstate*;

21845 switches the state (start condition) to *newstate*. If the string *newstate* has not been  
 21846 declared previously as a start condition in the *Definitions* section, the results are  
 21847 unspecified. The initial state is indicated by the digit '0' or the token **INITIAL**.

21848 The functions or macros described below are accessible to user code included in the *lex* input. It  
 21849 is unspecified whether they appear in the C code output of *lex*, or are accessible only through the  
 21850 **-ll** operand to *c89* or *cc* (the *lex* library).

21851 **int yylex(void)**

21852 Performs lexical analysis on the input; this is the primary function generated by the *lex*  
 21853 utility. The function shall return zero when the end of input is reached; otherwise, it shall  
 21854 return non-zero values (tokens) determined by the actions that are selected.

21855 **int yymore(void)**

21856 When called, indicates that when the next input string is recognized, it is to be appended to  
 21857 the current value of *yytext* rather than replacing it; the value in *yylen* shall be adjusted  
 21858 accordingly.

21859 **int yyless(int n)**

21860 Retains *n* initial characters in *yytext*, NUL-terminated, and treats the remaining characters  
 21861 as if they had not been read; the value in *yylen* shall be adjusted accordingly.

21862 **int input(void)**

21863 Returns the next character from the input, or zero on end-of-file. It shall obtain input from  
 21864 the stream pointer *yyin*, although possibly via an intermediate buffer. Thus, once scanning  
 21865 has begun, the effect of altering the value of *yyin* is undefined. The character read is  
 21866 removed from the input stream of the scanner without any processing by the scanner.

21867 **int unput(int c)**  
 21868 Returns the character 'c' to the input; *yytext* and *yylen* are undefined until the next  
 21869 expression is matched. The result of using *unput* for more characters than have been input is  
 21870 unspecified.

21871 The following functions appear only in the *lex* library accessible through the `-ll` operand; they  
 21872 can therefore be redefined by a portable application:

21873 **int yywrap(void)**  
 21874 Called by *yylex* at end-of-file; the default *yywrap* always shall return 1. If the application  
 21875 requires *yylex* to continue processing with another source of input, then the application can  
 21876 include a function *yywrap*, which associates another file with the external variable **FILE\****yyin*  
 21877 and shall return a value of zero.

21878 **int main(int argc, char \*argv[ ])**  
 21879 Calls *yylex* to perform lexical analysis, then exits. The user code can contain *main* to perform  
 21880 application-specific operations, calling *yylex* as applicable.

21881 Except for *input*, *unput*, and *main*, all external and static names generated by *lex* shall begin with  
 21882 the prefix **yy** or **YY**.

#### 21883 EXIT STATUS

21884 The following exit values shall be returned:

21885 0 Successful completion.

21886 >0 An error occurred.

#### 21887 CONSEQUENCES OF ERRORS

21888 Default.

#### 21889 APPLICATION USAGE

21890 Portable applications are warned that in the *Rules* section, an *ERE* without an action is not  
 21891 acceptable, but need not be detected as erroneous by *lex*. This may result in compilation or  
 21892 runtime errors.

21893 The purpose of *input* is to take characters off the input stream and discard them as far as the  
 21894 lexical analysis is concerned. A common use is to discard the body of a comment once the  
 21895 beginning of a comment is recognized.

21896 The *lex* utility is not fully internationalized in its treatment of regular expressions in the *lex*  
 21897 source code or generated lexical analyzer. It would seem desirable to have the lexical analyzer  
 21898 interpret the regular expressions given in the *lex* source according to the environment specified  
 21899 when the lexical analyzer is executed, but this is not possible with the current *lex* technology.  
 21900 Furthermore, the very nature of the lexical analyzers produced by *lex* must be closely tied to the  
 21901 lexical requirements of the input language being described, which is frequently locale-specific  
 21902 anyway. (For example, writing an analyzer that is used for French text is not automatically  
 21903 useful for processing other languages.)

#### 21904 EXAMPLES

21905 The following is an example of a *lex* program that implements a rudimentary scanner for a  
 21906 Pascal-like syntax:

```
21907 %{
21908 /* Need this for the call to atof() below. */
21909 #include <math.h>
21910 /* Need this for printf(), fopen(), and stdin below. */
21911 #include <stdio.h>
21912 %}
```

```

21913 DIGIT [0-9]
21914 ID [a-z][a-z0-9]*
21915 %%
21916 {DIGIT}+ {
21917 printf("An integer: %s (%d)\n", yytext,
21918 atoi(yytext));
21919 }
21920 {DIGIT}+"."{DIGIT}* {
21921 printf("A float: %s (%g)\n", yytext,
21922 atof(yytext));
21923 }
21924 if|then|begin|end|procedure|function {
21925 printf("A keyword: %s\n", yytext);
21926 }
21927 {ID} printf("An identifier: %s\n", yytext);
21928 "+"|"-"|"*"|"/" printf("An operator: %s\n", yytext);
21929 "{ "[^]\n]*" /* Eat up one-line comments. */
21930 [\t\n]+ /* Eat up white space. */
21931 . printf("Unrecognized character: %s\n", yytext);
21932 %%
21933 int main(int argc, char *argv[])
21934 {
21935 ++argv, --argc; /* Skip over program name. */
21936 if (argc > 0)
21937 yyin = fopen(argv[0], "r");
21938 else
21939 yyin = stdin;
21940 yylex();
21941 }

```

#### 21942 RATIONALE

21943 Even though the `-c` option and references to the C language are retained in this description, *lex*  
 21944 may be generalized to other languages, as was done at one time for EFL, the Extended  
 21945 FORTRAN Language. Since the *lex* input specification is essentially language-independent,  
 21946 versions of this utility could be written to produce Ada, Modula-2, or Pascal code, and there are  
 21947 known historical implementations that do so.

21948 The current description of *lex* bypasses the issue of dealing with internationalized EREs in the *lex*  
 21949 source code or generated lexical analyzer. If it follows the model used by *awk* (the source code is  
 21950 assumed to be presented in the POSIX locale, but input and output are in the locale specified by  
 21951 the environment variables), then the tables in the lexical analyzer produced by *lex* would  
 21952 interpret EREs specified in the *lex* source in terms of the environment variables specified when  
 21953 *lex* was executed. The desired effect would be to have the lexical analyzer interpret the EREs  
 21954 given in the *lex* source according to the environment specified when the lexical analyzer is  
 21955 executed, but this is not possible with the current *lex* technology.

21956 The description of octal and hexadecimal-digit escape sequences agrees with the ISO C standard  
 21957 use of escape sequences. See the RATIONALE for *ed* on page 369 for a discussion of bytes larger

- 21958 than 9 bits being represented by octal values. Hexadecimal values can represent larger bytes and  
21959 multi-byte characters directly, using as many digits as required.
- 21960 There is no detailed output format specification. The observed behavior of *lex* under four  
21961 different historical implementations was that none of these implementations consistently  
21962 reported the line numbers for error and warning messages. Furthermore, there was a desire that  
21963 *lex* be allowed to output additional diagnostic messages. Leaving message formats unspecified  
21964 avoids these formatting questions and problems with internationalization.
- 21965 Although the `%x` specifier for *exclusive* start conditions is not historical practice, it is believed to  
21966 be a minor change to historical implementations and greatly enhances the usability of *lex*  
21967 programs since it permits an application to obtain the expected functionality with fewer  
21968 statements.
- 21969 The `%array` and `%pointer` declarations were added as a compromise between historical systems.  
21970 The System V-based *lex* copies the matched text to a *yytext* array. The *flex* program, supported in  
21971 BSD and GNU systems, uses a pointer. In the latter case, significant performance improvements  
21972 are available for some scanners. Most historical programs should require no change in porting  
21973 from one system to another because the string being referenced is null-terminated in both cases.  
21974 (The method used by *flex* in its case is to null-terminate the token in place by remembering the  
21975 character that used to come right after the token and replacing it before continuing on to the next  
21976 scan.) Multi-file programs with external references to *yytext* outside the scanner source file  
21977 should continue to operate on their historical systems, but would require one of the new  
21978 declarations to be considered strictly portable.
- 21979 The description of EREs avoids unnecessary duplication of ERE details because their meanings  
21980 within a *lex* ERE are the same as that for the ERE in this volume of IEEE Std. 1003.1-200x.
- 21981 The reason for the undefined condition associated with text beginning with a <blank> or within  
21982 "`% {`" and "`% }`" delimiter lines appearing in the *Rules* section is historical practice. Both the BSD  
21983 and System V *lex* copy the indented (or enclosed) input in the *Rules* section (except at the  
21984 beginning) to unreachable areas of the *yylex* function (the code is written directly after a *break*  
21985 statement). In some cases, the System V *lex* generates an error message or a syntax error,  
21986 depending on the form of indented input.
- 21987 The intention in breaking the list of functions into those that may appear in *lex.yy.c* versus those  
21988 that only appear in *libl.a* is that only those functions in *libl.a* can be reliably redefined by a  
21989 portable application.
- 21990 The descriptions of standard output and standard error are somewhat complicated because  
21991 historical *lex* implementations chose to issue diagnostic messages to standard output (unless `-t`  
21992 was given). This standard allows this behavior, but leaves an opening for the more expected  
21993 behavior of using standard error for diagnostics. Also, the System V behavior of writing the  
21994 statistics when any table sizes are given is allowed, while BSD-derived systems can avoid it. The  
21995 programmer can always precisely obtain the desired results by using either the `-t` or `-n` options.
- 21996 The OPERANDS section does not mention the use of `-` as a synonym for standard input; not all  
21997 historical implementations support such usage for any of the *file* operands.
- 21998 A description of the *translation table* was deleted from early proposals because of its relatively  
21999 low usage in historical applications.
- 22000 The change to the definition of the *input* function that allows buffering of input presents the  
22001 opportunity for major performance gains in some applications.
- 22002 The following examples clarify the differences between *lex* regular expressions and regular  
22003 expressions appearing elsewhere in this volume of IEEE Std. 1003.1-200x. For regular  
22004 expressions of the form "`r/x`", the string matching *r* is always returned; confusion may arise

22005 when the beginning of *x* matches the trailing portion of *r*. For example, given the regular  
 22006 expression "*a\*b/cc*" and the input "*aaabcc*", *yytext* would contain the string "*aaab*" on this  
 22007 match. But given the regular expression "*x\*/xy*" and the input "*xxxxy*", the token **xxx**, not **xx**,  
 22008 is returned by some implementations because **xxx** matches "*x\**".

22009 In the rule "*ab\*/bc*", the "*b\**" at the end of *r* extends *r*'s match into the beginning of the  
 22010 trailing context, so the result is unspecified. If this rule were "*ab/bc*", however, the rule  
 22011 matches the text "*ab*" when it is followed by the text "*bc*". In this latter case, the matching of *r*  
 22012 cannot extend into the beginning of *x*, so the result is specified.

#### 22013 FUTURE DIRECTIONS

22014 None.

#### 22015 SEE ALSO

22016 *c89*, *yacc*

#### 22017 CHANGE HISTORY

22018 First released in Issue 2.

#### 22019 Issue 4

22020 Aligned with the ISO/IEC 9945-2:1993 standard.

#### 22021 Issue 6

22022 This utility is now marked as part of the C-Language Development Utilities option.

22023 The obsolescent **-c** option is withdrawn in this issue. |

22024 The normative text is reworded to avoid use of the term "must" for application requirements. |

22025 **NAME**22026       link — call *link()* function22027 **SYNOPSIS**22028 XSI     link *file1 file2*

22029

22030 **DESCRIPTION**22031       The *link* utility shall perform the function call:22032       link(*file1*, *file2*);22033       A user may need appropriate privilege to invoke the *link* utility.22034 **OPTIONS**

22035       None.

22036 **OPERANDS**

22037       The following operands shall be supported:

22038       *file1*       The path name of an existing file.22039       *file2*       The path name of the new directory entry to be created.22040 **STDIN**

22041       Not used.

22042 **INPUT FILES**

22043       Not used.

22044 **ENVIRONMENT VARIABLES**22045       The following environment variables shall affect the execution of *link*:

22046       *LANG*       Provide a default value for the internationalization variables that are unset or null.  
 22047       If *LANG* is unset or null, the corresponding value from the implementation-  
 22048       dependent default locale shall be used. If any of the internationalization variables  
 22049       contain an invalid setting, the utility behaves as if none of the variables had been  
 22050       set.

22051       *LC\_ALL*     If set to a non-empty string value, override the values of all the other  
 22052       internationalization variables.

22053       *LC\_CTYPE*   Determine the locale for the interpretation of sequences of bytes of text data as  
 22054       characters (for example, single-byte as opposed to multi-byte characters in  
 22055       arguments).

22056       *LC\_MESSAGES*  
 22057       Determine the locale that should be used to affect the format and contents of  
 22058       diagnostic messages written to standard error.

22059       *NLSPATH*    Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

22060 **ASYNCHRONOUS EVENTS**

22061       Default.

22062 **STDOUT**

22063       None.



**22064 STDERR**

22065           Used only for diagnostic messages.

**22066 OUTPUT FILES**

22067           None.

**22068 EXTENDED DESCRIPTION**

22069           None.

**22070 EXIT STATUS**

22071           The following exit values shall be returned:

22072           0   Successful completion.

22073           >0  An error occurred.

**22074 CONSEQUENCES OF ERRORS**

22075           Default.

**22076 APPLICATION USAGE**

22077           None.

**22078 EXAMPLES**

22079           None.

**22080 RATIONALE**

22081           None.

**22082 FUTURE DIRECTIONS**

22083           None.

**22084 SEE ALSO**

22085           *In, unlink*, the System Interfaces volume of IEEE Std. 1003.1-200x, *link()*

**22086 CHANGE HISTORY**

22087           First released in Issue 5.

22088 **NAME**

22089       ln — link files

22090 **SYNOPSIS**22091       ln [-fs] *source\_file target\_file* |22092       ln [-fs] *source\_file ... target\_dir* |22093 **DESCRIPTION**

22094       In the first synopsis form, the *ln* utility shall create a new directory entry (link), or if the *-s*  
 22095       option is specified a symbolic link, for the file specified by the *source\_file* operand, at the  
 22096       *destination* path specified by the *target\_file* operand. This first synopsis form shall be assumed  
 22097       when the final operand does not name an existing directory; if more than two operands are  
 22098       specified and the final is not an existing directory, an error shall result.

22099       In the second synopsis form, the *ln* utility shall create a new directory entry (link), or if the *-s*  
 22100       option is specified a symbolic link, for each file specified by a *source\_file* operand, at a *destination*  
 22101       path in the existing directory named by *target\_dir*.

22102       If the last operand specifies an existing file of a type not specified by the System Interfaces  
 22103       volume of IEEE Std. 1003.1-200x, the behavior is implementation-dependent.

22104       The corresponding *destination* path for each *source\_file* shall be the concatenation of the target  
 22105       directory path name, a slash character, and the last path name component of the *source\_file*. The  
 22106       second synopsis form shall be assumed when the final operand names an existing directory.

22107       For each *source\_file*:

- 22108       1. If the *destination* path exists:
  - 22109           a. If the *-f* option is not specified, *ln* shall write a diagnostic message to standard error,  
 22110           do nothing more with the current *source\_file*, and go on to any remaining *source\_files*.
  - 22111           b. Actions shall be performed equivalent to the *unlink()* function defined in the System  
 22112           Interfaces volume of IEEE Std. 1003.1-200x, called using *destination* as the *path*  
 22113           argument. If this fails for any reason, *ln* shall write a diagnostic message to standard  
 22114           error, do nothing more with the current *source\_file*, and go on to any remaining  
 22115           *source\_files*.
- 22116       2. If the *-s* option is specified, *ln* shall create a symbolic link named by the *destination* path  
 22117       and containing as its path name *source\_file*. The *ln* utility shall do nothing more with  
 22118       *source\_file* and shall go on to any remaining files.
- 22119       3. If *source\_file* is a symbolic link, actions shall be performed equivalent to the *link()* function  
 22120       using the object that *source\_file* references as the *path1* argument and the destination path  
 22121       as the *path2* argument. The *ln* utility shall do nothing more with *source\_file* and shall go on  
 22122       to any remaining files.
- 22123       4. Actions shall be performed equivalent to the *link()* function defined in the System  
 22124       Interfaces volume of IEEE Std. 1003.1-200x using *source\_file* as the *path1* argument, and the  
 22125       *destination* path as the *path2* argument.

22126 **OPTIONS**

22127       The *ln* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 22128       Section 12.2, Utility Syntax Guidelines. |

22129       The following option shall be supported:

22130       *-f*           Force existing *destination* path names to be removed to allow the link. |

|       |                                                                               |                                                                                                                                                                                                                                                                                                                                                                  |  |
|-------|-------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| 22131 | <b>-s</b>                                                                     | Create symbolic links instead of hard links.                                                                                                                                                                                                                                                                                                                     |  |
| 22132 | <b>OPERANDS</b>                                                               |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22133 | The following operands shall be supported:                                    |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22134 | <i>source_file</i>                                                            | A path name of a file to be linked. If the <b>-s</b> option is specified, no restrictions on the type of file or on its existence shall be made. If the <b>-s</b> option is not specified, whether a directory can be linked is implementation-dependent.                                                                                                        |  |
| 22135 |                                                                               |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22136 |                                                                               |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22137 | <i>target_file</i>                                                            | The path name of the new directory entry to be created.                                                                                                                                                                                                                                                                                                          |  |
| 22138 | <i>target_dir</i>                                                             | A path name of an existing directory in which the new directory entries are created.                                                                                                                                                                                                                                                                             |  |
| 22139 |                                                                               |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22140 | <b>STDIN</b>                                                                  |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22141 | Not used.                                                                     |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22142 | <b>INPUT FILES</b>                                                            |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22143 | None.                                                                         |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22144 | <b>ENVIRONMENT VARIABLES</b>                                                  |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22145 | The following environment variables shall affect the execution of <i>ln</i> : |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22146 | <i>LANG</i>                                                                   | Provide a default value for the internationalization variables that are unset or null. If <i>LANG</i> is unset or null, the corresponding value from the implementation-dependent default locale shall be used. If any of the internationalization variables contains an invalid setting, the utility shall behave as if none of the variables had been defined. |  |
| 22147 |                                                                               |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22148 |                                                                               |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22149 |                                                                               |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22150 |                                                                               |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22151 | <i>LC_ALL</i>                                                                 | If set to a non-empty string value, override the values of all the other internationalization variables.                                                                                                                                                                                                                                                         |  |
| 22152 |                                                                               |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22153 | <i>LC_CTYPE</i>                                                               | Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).                                                                                                                                                                                        |  |
| 22154 |                                                                               |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22155 |                                                                               |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22156 | <i>LC_MESSAGES</i>                                                            |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22157 |                                                                               | Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.                                                                                                                                                                                                                                     |  |
| 22158 |                                                                               |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22159 | XSI <i>NLSPATH</i>                                                            | Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .                                                                                                                                                                                                                                                                            |  |
| 22160 | <b>ASYNCHRONOUS EVENTS</b>                                                    |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22161 | Default.                                                                      |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22162 | <b>STDOUT</b>                                                                 |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22163 | Not used.                                                                     |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22164 | <b>STDERR</b>                                                                 |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22165 | Used only for diagnostic messages.                                            |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22166 | <b>OUTPUT FILES</b>                                                           |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22167 | None.                                                                         |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22168 | <b>EXTENDED DESCRIPTION</b>                                                   |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22169 | None.                                                                         |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22170 | <b>EXIT STATUS</b>                                                            |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22171 | The following exit values shall be returned:                                  |                                                                                                                                                                                                                                                                                                                                                                  |  |
| 22172 | 0                                                                             | All the specified files were linked successfully.                                                                                                                                                                                                                                                                                                                |  |

22173 >0 An error occurred.

## 22174 CONSEQUENCES OF ERRORS

22175 Default.

## 22176 APPLICATION USAGE

22177 None.

## 22178 EXAMPLES

22179 None.

## 22180 RATIONALE

22181 Some historic versions of *ln* (including the one specified by the SVID, unlink the destination file,  
22182 if it exists, by default. If the mode does not permit writing, these versions prompt for  
22183 confirmation before attempting the unlink. In these versions the *-f* option causes *ln* not to  
22184 attempt to prompt for confirmation.

22185 This allows *ln* to succeed in creating links when the target file already exists, even if the file itself  
22186 is not writable (although the directory must be). Early proposals specified this functionality.

22187 This volume of IEEE Std. 1003.1-200x does not allow the *ln* utility to unlink existing destination  
22188 paths by default for the following reasons:

- 22189 • The *ln* utility has historically been used to provide locking for shell applications, a usage that  
22190 is incompatible with *ln* unlinking the destination path by default. There was no  
22191 corresponding technical advantage to adding this functionality.
- 22192 • This functionality gave *ln* the ability to destroy the link structure of files, which changes the  
22193 historical behavior of *ln*.
- 22194 • This functionality is easily replicated with a combination of *rm* and *ln*.
- 22195 • It is not historical practice in many systems; BSD and BSD-derived systems do not support  
22196 this behavior. Unfortunately, whichever behavior is selected can cause scripts written  
22197 expecting the other behavior to fail.
- 22198 • It is preferable that *ln* perform in the same manner as the *link()* function, which does not  
22199 permit the target to exist already.

22200 This volume of IEEE Std. 1003.1-200x retains the *-f* option to provide support for shell scripts  
22201 depending on the SVID semantics. It seems likely that shell scripts would not be written to  
22202 handle prompting by *ln* and would therefore have specified the *-f* option.

22203 The *-f* option is an undocumented feature of many historical versions of the *ln* utility, allowing  
22204 linking to directories. These versions require modification.

22205 Early proposals of this volume of IEEE Std. 1003.1-200x also required an *-i* option, which  
22206 behaved like the *-i* options in *cp* and *mv*, prompting for confirmation before unlinking existing  
22207 files. This was not historical practice for the *ln* utility and has been omitted.

## 22208 FUTURE DIRECTIONS

22209 None.

## 22210 SEE ALSO

22211 *chmod*, *find*, *pax*, *rm*, the System Interfaces volume of IEEE Std. 1003.1-200x, *link()*

## 22212 CHANGE HISTORY

22213 First released in Issue 2.

22214 **Issue 4**

22215 Aligned with the ISO/IEC 9945-2:1993 standard.

22216 **Issue 6**22217 The *ln* utility is updated to include symbolic link processing as defined in the IEEE P1003.2b  
22218 draft standard.

22219 **NAME**

22220 locale — get locale-specific information

22221 **SYNOPSIS**

22222 locale [-a | -m]

22223 locale [-ck] *name*...22224 **DESCRIPTION**

22225 The *locale* utility shall write information about the current locale environment, or all public  
 22226 locales, to the standard output. For the purposes of this section, a *public locale* is one provided by  
 22227 the implementation that is accessible to the application.

22228 When *locale* is invoked without any arguments, it shall summarize the current locale  
 22229 environment for each locale category as determined by the settings of the environment variables  
 22230 defined in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 7, Locale.

22231 When invoked with operands, it shall write values that have been assigned to the keywords in  
 22232 the locale categories, as follows:

- 22233 • Specifying a keyword name shall select the named keyword and the category containing that  
 22234 keyword.
- 22235 • Specifying a category name shall select the named category and all keywords in that  
 22236 category.

22237 **OPTIONS**

22238 The *locale* utility shall conform to the System Interface Definitions volume of  
 22239 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

22240 The following options shall be supported:

- 22241 **-a** Write information about all available public locales. The available locales shall  
 22242 include **POSIX**, representing the POSIX locale. The manner in which the  
 22243 implementation determines what other locales are available is implementation-  
 22244 dependent.
- 22245 **-c** Write the names of selected locale categories; see the STDOUT section. The **-c**  
 22246 option increases readability when more than one category is selected (for example,  
 22247 via more than one keyword name or via a category name). It is valid both with  
 22248 and without the **-k** option.
- 22249 **-k** Write the names and values of selected keywords. The implementation may omit  
 22250 values for some keywords; see the OPERANDS section.
- 22251 **-m** Write names of available charmaps; see the System Interface Definitions volume of  
 22252 IEEE Std. 1003.1-200x, Section 6.1, Portable Character Set.

22253 **OPERANDS**

22254 The following operand shall be supported:

- 22255 *name* The name of a locale category as defined in the System Interface Definitions  
 22256 volume of IEEE Std. 1003.1-200x, Chapter 7, Locale, the name of a keyword in a  
 22257 locale category, or the reserved name **charmap**. The named category or keyword  
 22258 shall be selected for output. If a single *name* represents both a locale category name  
 22259 and a keyword name in the current locale, the results are unspecified. Otherwise,  
 22260 both category and keyword names can be specified as *name* operands, in any  
 22261 sequence. It is implementation-dependent whether any keyword values are  
 22262 written for the categories *LC\_CTYPE* and *LC\_COLLATE*.

22263 **STDIN**

22264 Not used.

22265 **INPUT FILES**

22266 None.

22267 **ENVIRONMENT VARIABLES**22268 The following environment variables shall affect the execution of *locale*:

22269 **LANG** Provide a default value for the internationalization variables that are unset or null.  
 22270 If *LANG* is unset or null, the corresponding value from the implementation-  
 22271 dependent default locale shall be used. If any of the internationalization variables  
 22272 contains an invalid setting, the utility shall behave as if none of the variables had  
 22273 been defined.

22274 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 22275 internationalization variables.

22276 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 22277 characters (for example, single-byte as opposed to multi-byte characters in  
 22278 arguments and input files).

22279 **LC\_MESSAGES**

22280 Determine the locale that should be used to affect the format and contents of  
 22281 diagnostic messages written to standard error.

22282 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

22283 XSI The application shall ensure that the *LANG*, *LC\_\**, and *NLSPATH* environment variables specify  
 22284 the current locale environment to be written out; they shall be used if the *-a* option is not  
 22285 specified.

22286 **ASYNCHRONOUS EVENTS**

22287 Default.

22288 **STDOUT**

22289 If *locale* is invoked without any options or operands, the names and values of the *LANG* and  
 22290 *LC\_\** environment variables described in this volume of IEEE Std. 1003.1-200x shall be written to  
 22291 the standard output, one variable per line, with *LANG* first, and each line using the following  
 22292 format. Only those variables set in the environment and not overridden by *LC\_ALL* shall be  
 22293 written using this format:

22294 "%s=%s\n", <variable\_name>, <value>

22295 The names of those *LC\_\** variables associated with locale categories defined in this volume of  
 22296 IEEE Std. 1003.1-200x that are not set in the environment or are overridden by *LC\_ALL* shall be  
 22297 written in the following format:

22298 "%s=\" \"%s\" \"\n", <variable\_name>, <implied value>

22299 The <implied value> shall be the name of the locale that has been selected for that category by the  
 22300 implementation, based on the values in *LANG* and *LC\_ALL*, as described in the System Interface  
 22301 Definitions volume of IEEE Std. 1003.1-200x, Chapter 8, Environment Variables.

22302 The <value> and <implied value> shown above shall be properly quoted for possible later reentry  
 22303 to the shell. The <value> shall not be quoted using double-quotes (so that it can be distinguished  
 22304 by the user from the <implied value> case, which always requires double-quotes).

22305 The *LC\_ALL* variable shall be written last, using the first format shown above. If it is not set, it  
 22306 shall be written as:

22307 "LC\_ALL=\n"

22308 If any arguments are specified:

22309 1. If the **-a** option is specified, the names of all the public locales shall be written, each in the  
22310 following format:

22311 "%s\n", <locale name>

22312 2. If the **-c** option is specified, the names of all selected categories shall be written, each in the  
22313 following format:

22314 "%s\n", <category name>

22315 If keywords are also selected for writing (see following items), the category name output  
22316 shall precede the keyword output for that category.

22317 If the **-c** option is not specified, the names of the categories shall not be written; only the  
22318 keywords, as selected by the <name> operand, shall be written.

22319 3. If the **-k** option is specified, the names and values of selected keywords shall be written. If  
22320 a value is non-numeric, it shall be written in the following format:

22321 "%s=\"%s\"\\n", <keyword name>, <keyword value>

22322 If the keyword was **charmap**, the name of the charmap (if any) that was specified via the  
22323 *localedef* **-f** option when the locale was created shall be written, with the word **charmap** as  
22324 <keyword name>.

22325 If a value is numeric, it shall be written in one of the following formats:

22326 "%s=%d\n", <keyword name>, <keyword value>

22327 "%s=%c%o\n", <keyword name>, <escape character>, <keyword value>

22328 "%s=%cx%x\n", <keyword name>, <escape character>, <keyword value>

22329 where the <escape character> is that identified by the **escape\_char** keyword in the current  
22330 locale; see the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 7.3,  
22331 Locale Definition.

22332 Compound keyword values (list entries) shall be separated in the output by semicolons.  
22333 When included in keyword values, the semicolon, the double-quote, the backslash, and  
22334 any control character shall be preceded (escaped) with the escape character.

22335 4. If the **-k** option is not specified, selected keyword values shall be written, each in the  
22336 following format:

22337 "%s\n", <keyword value>

22338 If the keyword was **charmap**, the name of the charmap (if any) that was specified via the  
22339 *localedef* **-f** option when the locale was created shall be written.

22340 5. If the **-m** option is specified, then a list of all available charmaps shall be written, each in  
22341 the format:

22342 "%s\n", <charmap>

22343 where <charmap> is in a format suitable for use as the option-argument to the *localedef* **-f**  
22344 option.



22345 **STDERR**

22346       Used only for diagnostic messages.

22347 **OUTPUT FILES**

22348       None.

22349 **EXTENDED DESCRIPTION**

22350       None.

22351 **EXIT STATUS**

22352       The following exit values shall be returned:

22353       0   All the requested information was found and output successfully.

22354       &gt;0   An error occurred.

22355 **CONSEQUENCES OF ERRORS**

22356       Default.

22357 **APPLICATION USAGE**

22358       If the *LANG* environment variable is not set or set to an empty value, or one of the *LC\_\**  
 22359       environment variables is set to an unrecognized value, the actual locales assumed (if any) are  
 22360       implementation-dependent as described in the System Interface Definitions volume of  
 22361       IEEE Std. 1003.1-200x, Chapter 8, Environment Variables.

22362       Implementations are not required to write out the actual values for keywords in the categories  
 22363       *LC\_CTYPE* and *LC\_COLLATE*; however, they must write out the categories (allowing an  
 22364       application to determine, for example, which character classes are available).

22365 **EXAMPLES**

22366       In the following examples, the assumption is that locale environment variables are set as  
 22367       follows:

22368       *LANG*=locale\_x  
 22369       *LC\_COLLATE*=locale\_y

22370       The command *locale* would result in the following output:

22371       *LANG*=locale\_x  
 22372       *LC\_CTYPE*="locale\_x"  
 22373       *LC\_COLLATE*=locale\_y  
 22374       *LC\_TIME*="locale\_x"  
 22375       *LC\_NUMERIC*="locale\_x"  
 22376       *LC\_MONETARY*="locale\_x"  
 22377       *LC\_MESSAGES*="locale\_x"  
 22378       *LC\_ALL*=

22379       The order of presentation of the categories is not specified by this volume of  
 22380       IEEE Std. 1003.1-200x.

22381       The command:

22382       *LC\_ALL*=POSIX *locale* -ck decimal\_point

22383       would produce:

22384       *LC\_NUMERIC*  
 22385       decimal\_point="."

22386       The following command shows an application of *locale* to determine whether a user-supplied  
 22387       response is affirmative:

```

22388 if printf "%s\n" "$response" | grep -Eq "$(locale yesexpr)"
22389 then
22390 affirmative processing goes here
22391 else
22392 non-affirmative processing goes here
22393 fi

```

#### 22394 RATIONALE

22395 The output for categories *LC\_CTYPE* and *LC\_COLLATE* has been made implementation-  
 22396 dependent because there is a questionable value in having a shell script receive an entire array of  
 22397 characters. It is also difficult to return a logical collation description, short of returning a  
 22398 complete *localedef* source.

22399 The **-m** option was included to allow applications to query for the existence of charmaps. The  
 22400 output is a list of the charmaps (implementation-supplied and user-supplied, if any) on the  
 22401 system.

22402 The **-c** option was included for readability when more than one category is selected (for  
 22403 example, via more than one keyword name or via a category name). It is valid both with and  
 22404 without the **-k** option.

22405 The **charmap** keyword, which returns the name of the charmap (if any) that was used when the  
 22406 current locale was created, was included to allow applications needing the information to  
 22407 retrieve it.

#### 22408 FUTURE DIRECTIONS

22409 None.

#### 22410 SEE ALSO

22411 *localedef*, the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 7.3, Locale  
 22412 Definition

#### 22413 CHANGE HISTORY

22414 First released in Issue 4.

#### 22415 Issue 5

22416 FUTURE DIRECTIONS section added.

#### 22417 Issue 6

22418 The normative text is reworded to avoid use of the term “must” for application requirements.

22419 **NAME**

22420 localedef — define locale environment

22421 **SYNOPSIS**22422 localedef [-c][-f *charmap*][-i *sourcefile*][-u *code\_set\_name*] *name*22423 **DESCRIPTION**

22424 The *localedef* utility shall convert source definitions for locale categories into a format usable by  
 22425 the functions and utilities whose operational behavior is determined by the setting of the locale  
 22426 environment variables defined in the System Interface Definitions volume of  
 22427 IEEE Std. 1003.1-200x, Chapter 7, Locale. It is implementation-dependent whether users have the  
 22428 capability to create new locales, in addition to those supplied by the implementation. If the  
 22429 symbolic constant POSIX2\_LOCALEDEF is defined, the system supports the creation of new  
 22430 XSI locales. On XSI-conformant systems, the symbolic constant POSIX2\_LOCALEDEF shall be  
 22431 defined.

22432 The utility shall read source definitions for one or more locale categories belonging to the same  
 22433 locale from the file named in the -i option (if specified) or from standard input.

22434 The *name* operand identifies the target locale. The utility shall support the creation of *public*, or  
 22435 generally accessible locales, as well as *private*, or restricted-access locales. Implementations may  
 22436 restrict the capability to create or modify public locales to users with the appropriate privileges.

22437 Each category source definition shall be identified by the corresponding environment variable  
 22438 name and terminated by an **END category-name** statement. The following categories shall be  
 22439 supported. In addition, the input may contain source for implementation-dependent categories.

22440 LC\_CTYPE Defines character classification and case conversion.

22441 LC\_COLLATE

22442 Defines collation rules.

22443 LC\_MONETARY

22444 Defines the format and symbols used in formatting of monetary information.

22445 LC\_NUMERIC

22446 Defines the decimal delimiter, grouping, and grouping symbol for non-monetary  
 22447 numeric editing.

22448 LC\_TIME Defines the format and content of date and time information.

22449 LC\_MESSAGES

22450 Defines the format and values of affirmative and negative responses.

22451 **OPTIONS**

22452 The *localedef* utility shall conform to the System Interface Definitions volume of  
 22453 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

22454 The following options shall be supported:

22455 -c Create permanent output even if warning messages have been issued.

22456 -f *charmap* Specify the path name of a file containing a mapping of character symbols and  
 22457 collating element symbols to actual character encodings. The format of the  
 22458 *charmap* is described under the System Interface Definitions volume of  
 22459 IEEE Std. 1003.1-200x, Section 6.4, Character Set Description File. The application  
 22460 shall ensure that this option is specified if symbolic names (other than collating  
 22461 symbols defined in a **collating-symbol** keyword) are used. If the -f option is not  
 22462 present, an implementation-dependent character mapping shall be used.

22463        **-i inputfile**   The path name of a file containing the source definitions. If this option is not  
22464                           present, source definitions shall be read from standard input. The format of the  
22465                           *inputfile* is described in the System Interface Definitions volume of  
22466                           IEEE Std. 1003.1-200x, Section 7.3, Locale Definition.

22467        **-u code\_set\_name** Specify the name of a codeset used as the target mapping of character symbols  
22468                           and collating element symbols whose encoding values are defined in terms of the  
22469                           ISO/IEC 10646-1: 1993 standard position constant values.

## 22470 OPERANDS

22471        The following operand shall be supported:

22472        **name**           Identifies the locale; see the System Interface Definitions volume of  
22473                           IEEE Std. 1003.1-200x, Chapter 7, Locale for a description of the use of this name. If  
22474                           the name contains one or more slash characters, *name* shall be interpreted as a path  
22475                           name where the created locale definitions shall be stored. If *name* does not contain  
22476                           any slash characters, the interpretation of the name is implementation-dependent  
22477                           and the locale shall be public. This capability may be restricted to users with  
22478                           appropriate privileges. (As a consequence of specifying one *name*, although several  
22479                           categories can be processed in one execution, only categories belonging to the  
22480                           same locale can be processed.)

## 22481 STDIN

22482        Unless the **-i** option is specified, the standard input shall be a text file containing one or more  
22483        locale category source definitions, as described in the System Interface Definitions volume of  
22484        IEEE Std. 1003.1-200x, Section 7.3, Locale Definition. When lines are continued using the escape  
22485        character mechanism, there is no limit to the length of the accumulated continued line.

## 22486 INPUT FILES

22487        The character set mapping file specified as the *charmap* option-argument is described under the  
22488        System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 6.4, Character Set  
22489        Description File. If a locale category source definition contains a **copy** statement, as defined in  
22490        the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 7, Locale, and the  
22491        **copy** statement names a valid, existing locale, then *localedef* shall behave as if the source  
22492        definition had contained a valid category source definition for the named locale.

## 22493 ENVIRONMENT VARIABLES

22494        The following environment variables shall affect the execution of *localedef*:

22495        **LANG**           Provide a default value for the internationalization variables that are unset or null.  
22496                           If *LANG* is unset or null, the corresponding value from the implementation-  
22497                           dependent default locale shall be used. If any of the internationalization variables  
22498                           contains an invalid setting, the utility shall behave as if none of the variables had  
22499                           been defined.

22500        **LC\_ALL**        If set to a non-empty string value, override the values of all the other  
22501                           internationalization variables.

22502        **LC\_COLLATE**  
22503                           (This variable has no affect on *localedef*; the POSIX locale is used for this category.)

22504        **LC\_CTYPE**     Determine the locale for the interpretation of sequences of bytes of text data as  
22505                           characters (for example, single-byte as opposed to multi-byte characters in  
22506                           arguments and input files). This variable has no affect on the processing of *localedef*  
22507                           input data; the POSIX locale is used for this purpose, regardless of the value of this  
22508                           variable.

22509 **LC\_MESSAGES**  
 22510 Determine the locale that should be used to affect the format and contents of  
 22511 diagnostic messages written to standard error.

22512 XSI **NLSPATH** Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

22513 **ASYNCHRONOUS EVENTS**  
 22514 Default.

22515 **STDOUT**  
 22516 The utility shall report all categories successfully processed, in an unspecified format.

22517 **STDERR**  
 22518 Used only for diagnostic messages.

22519 **OUTPUT FILES**  
 22520 The format of the created output is unspecified. If the *name* operand does not contain a slash, the  
 22521 existence of an output file for the locale is unspecified.

22522 **EXTENDED DESCRIPTION**  
 22523 When the **-u** option is used, the *code\_set\_name* option-argument shall be interpreted as an  
 22524 implementation-dependent name of a codeset to which the ISO/IEC 10646-1:1993 standard  
 22525 position constant values shall be converted via an implementation-dependent method. Both the  
 22526 ISO/IEC 10646-1:1993 standard position constant values and other formats (decimal,  
 22527 hexadecimal, or octal) shall be valid as encoding values within the *charmap* file. The codeset  
 22528 represented by the implementation-dependent name can be any codeset that is supported by the  
 22529 implementation.

22530 When conflicts occur between the *charmap* specification of *<code\_set\_name>*, *<mb\_cur\_max>*, or  
 22531 *<mb\_cur\_min>* and the implementation-dependent interpretation of these respective items for  
 22532 the codeset represented by the **-u** option-argument *code\_set\_name*, the result is unspecified.

22533 When conflicts occur between the *charmap* encoding values specified for symbolic names of  
 22534 characters of the portable character set and the implementation-dependent assignment of  
 22535 character encoding values, the result is unspecified.

22536 If a non-printable character in the *charmap* has a width specified that is not **-1**, *localedef* shall  
 22537 generate a warning.

22538 **EXIT STATUS**  
 22539 The following exit values shall be returned:

22540 0 No errors occurred and the locales were successfully created.

22541 1 Warnings occurred and the locales were successfully created.

22542 2 The locale specification exceeded implementation limits or the coded character set or sets  
 22543 used were not supported by the implementation, and no locale was created.

22544 3 The capability to create new locales is not supported by the implementation.

22545 >3 Warnings or errors occurred and no output was created.

22546 **CONSEQUENCES OF ERRORS**  
 22547 If an error is detected, no permanent output shall be created.

22548 If warnings occur, permanent output shall be created if the **-c** option was specified. The  
 22549 following conditions shall cause warning messages to be issued:

22550 • If a symbolic name not found in the *charmap* file is used for the descriptions of the **LC\_CTYPE**  
 22551 or **LC\_COLLATE** categories (for other categories, this shall be an error condition).

- 22552 • If the number of operands to the **order** keyword exceeds the {COLL\_WEIGHTS\_MAX} limit.
- 22553 • If optional keywords not supported by the implementation are present in the source.
- 22554 • If a non-printable character has a width specified other than -1.
- 22555 Other implementation-dependent conditions may also cause warnings.

#### 22556 APPLICATION USAGE

22557 The *charmap* definition is optional, and is contained outside the locale definition. This allows  
 22558 both completely self-defined source files, and generic sources (applicable to more than one  
 22559 codeset). To aid portability, all *charmap* definitions must use the same symbolic names for the  
 22560 portable character set. As explained in the System Interface Definitions volume of  
 22561 IEEE Std. 1003.1-200x, Section 6.4, Character Set Description File, it is implementation-  
 22562 dependent whether or not users or applications can provide additional character set description  
 22563 files. Therefore, the -f option might be operable only when an implementation-dependent  
 22564 *charmap* is named.

#### 22565 EXAMPLES

22566 None.

#### 22567 RATIONALE

22568 The output produced by the *localedef* utility is implementation-dependent. The *name* operand is  
 22569 used to identify the specific locale. (As a consequence, although several categories can be  
 22570 processed in one execution, only categories belonging to the same locale can be processed.)

#### 22571 FUTURE DIRECTIONS

22572 None.

#### 22573 SEE ALSO

22574 *locale*, the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 7.3, Locale  
 22575 Definition

#### 22576 CHANGE HISTORY

22577 First released in Issue 4.

#### 22578 Issue 6

22579 The -u option is added, as specified in the IEEE P1003.2b draft standard.

22580 The normative text is reworded to avoid use of the term “must” for application requirements.

22581 **NAME**

22582       logger — log messages

22583 **SYNOPSIS**22584       logger *string* ...22585 **DESCRIPTION**

22586       The *logger* utility saves a message, in an unspecified manner and format, containing the *string*  
 22587       operands provided by the user. The messages are expected to be evaluated later by personnel  
 22588       performing system administration tasks.

22589       It is implementation-dependent whether messages written in locales other than the POSIX locale  
 22590       are effective.

22591 **OPTIONS**

22592       None.

22593 **OPERANDS**

22594       The following operand shall be supported:

22595       *string*       One of the string arguments whose contents are concatenated together, in the  
 22596       order specified, separated by single <space> characters.

22597 **STDIN**

22598       Not used.

22599 **INPUT FILES**

22600       None.

22601 **ENVIRONMENT VARIABLES**22602       The following environment variables shall affect the execution of *logger*:

22603       *LANG*       Provide a default value for the internationalization variables that are unset or null.  
 22604       If *LANG* is unset or null, the corresponding value from the implementation-  
 22605       dependent default locale shall be used. If any of the internationalization variables  
 22606       contains an invalid setting, the utility shall behave as if none of the variables had  
 22607       been defined.

22608       *LC\_ALL*      If set to a non-empty string value, override the values of all the other  
 22609       internationalization variables.

22610       *LC\_CTYPE*   Determine the locale for the interpretation of sequences of bytes of text data as  
 22611       characters (for example, single-byte as opposed to multi-byte characters in  
 22612       arguments).

22613       *LC\_MESSAGES*

22614       Determine the locale that should be used to affect the format and contents of  
 22615       diagnostic messages written to standard error. (This means diagnostics from *logger*  
 22616       to the user or application, not diagnostic messages that the user is sending to the  
 22617       system administrator.)

22618 *XSI*       *NLSPATH*   Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

22619 **ASYNCHRONOUS EVENTS**

22620       Default.

22621 **STDOUT**

22622       Not used.

22623 **STDERR**

22624           Used only for diagnostic messages.

22625 **OUTPUT FILES**

22626           Unspecified.

22627 **EXTENDED DESCRIPTION**

22628           None.

22629 **EXIT STATUS**

22630           The following exit values shall be returned:

22631           0   Successful completion.

22632           >0  An error occurred.

22633 **CONSEQUENCES OF ERRORS**

22634           Default.

22635 **APPLICATION USAGE**

22636           This utility allows logging of information for later use by a system administrator or programmer  
22637           in determining why non-interactive utilities have failed. The locations of the saved messages,  
22638           their format, and retention period are all unspecified. There is no method for a portable  
22639           application to read messages, once written.

22640 **EXAMPLES**

22641           A batch application, running non-interactively, tries to read a configuration file and fails; it may  
22642           attempt to notify the system administrator with:

22643           logger myname: unable to read file foo. [timestamp]

22644 **RATIONALE**

22645           The standard developers believed strongly that some method of alerting administrators to errors  
22646           was necessary. The obvious example is a batch utility, running non-interactively, that is unable  
22647           to read its configuration files or that is unable to create or write its results file. However, the  
22648           standard developers did not wish to define the format or delivery mechanisms as they have  
22649           historically been (and will probably continue to be) very system-specific, as well as involving  
22650           functionality clearly outside of the scope of this volume of IEEE Std. 1003.1-200x.

22651           The text with *LC\_MESSAGES* about diagnostic messages means diagnostics from *logger* to the  
22652           user or application, not diagnostic messages that the user is sending to the system administrator.

22653           Multiple *string* arguments are allowed, similar to *echo*, for ease-of-use.

22654           Like the utilities *mailx* and *lp*, *logger* is admittedly difficult to test. This was not deemed sufficient  
22655           justification to exclude these utilities from this volume of IEEE Std. 1003.1-200x. It is also  
22656           arguable that they are, in fact, testable, but that the tests themselves are not portable.

22657 **FUTURE DIRECTIONS**

22658           None.

22659 **SEE ALSO**

22660           *mailx*, *write*

22661 **CHANGE HISTORY**

22662           First released in Issue 4.



22663 **NAME**

22664 logname — return the user's login name

22665 **SYNOPSIS**

22666 logname

22667 **DESCRIPTION**

22668 The *logname* utility shall write the user's login name to standard output. The login name shall be  
 22669 the string that would be returned by the *getlogin()* function defined in the System Interfaces  
 22670 volume of IEEE Std. 1003.1-200x. Under the conditions where the *getlogin()* function would fail,  
 22671 the *logname* utility shall write a diagnostic message to standard error and exit with a non-zero  
 22672 exit status.

22673 **OPTIONS**

22674 None.

22675 **OPERANDS**

22676 None.

22677 **STDIN**

22678 Not used.

22679 **INPUT FILES**

22680 None.

22681 **ENVIRONMENT VARIABLES**22682 The following environment variables shall affect the execution of *logname*:

22683 **LANG** Provide a default value for the internationalization variables that are unset or null.  
 22684 If *LANG* is unset or null, the corresponding value from the implementation-  
 22685 dependent default locale shall be used. If any of the internationalization variables  
 22686 contains an invalid setting, the utility shall behave as if none of the variables had  
 22687 been defined.

22688 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 22689 internationalization variables.

22690 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 22691 characters (for example, single-byte as opposed to multi-byte characters in  
 22692 arguments).

22693 **LC\_MESSAGES**

22694 Determine the locale that should be used to affect the format and contents of  
 22695 diagnostic messages written to standard error.

22696 **XS1** **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

22697 **ASYNCHRONOUS EVENTS**

22698 Default.

22699 **STDOUT**22700 The *logname* utility output shall be a single line consisting of the user's login name:

22701 "%s\n", &lt;login name&gt;

22702 **STDERR**

22703 Used only for diagnostic messages.

22704 **OUTPUT FILES**

22705           None.

22706 **EXTENDED DESCRIPTION**

22707           None.

22708 **EXIT STATUS**

22709           The following exit values shall be returned:

22710           0   Successful completion.

22711           &gt;0  An error occurred.

22712 **CONSEQUENCES OF ERRORS**

22713           Default.

22714 **APPLICATION USAGE**22715           The *logname* utility explicitly ignores the *LOGNAME* environment variable because environment changes could produce erroneous results.22717 **EXAMPLES**

22718           None.

22719 **RATIONALE**22720           The **passwd** file is not listed as required because the implementation may have other means of mapping login names.22722 **FUTURE DIRECTIONS**

22723           None.

22724 **SEE ALSO**22725           *id*, *who*22726 **CHANGE HISTORY**

22727           First released in Issue 2.

22728 **Issue 4**

22729           Aligned with the ISO/IEC 9945-2: 1993 standard.

## 22730 NAME

22731 lp — send files to a printer

## 22732 SYNOPSIS

22733 MAN lp [-c][-d *dest*][-n *copies*][-msw][-o *option*]... [-t *title*] [*file*...]

## 22734 DESCRIPTION

22735 The *lp* utility shall copy the input files to an output destination in an unspecified manner. The  
 22736 default output destination should be to a hardcopy device, such as a printer or microfilm  
 22737 recorder, that produces non-volatile, human-readable documents. If such a device is not  
 22738 available to the application, or if the system provides no such device, the *lp* utility shall exit with  
 22739 a non-zero exit status.

22740 The actual writing to the output device may occur some time after the *lp* utility successfully  
 22741 exits. During the portion of the writing that corresponds to each input file, the implementation  
 22742 shall guarantee exclusive access to the device.

22743 MAN The *lp* utility shall associate a unique *request ID* with each request.

22744 Normally, a banner page is produced to separate and identify each print job. This page may be  
 22745 suppressed by implementation-dependent conditions, such as an operator command or one of  
 22746 the *-o option* values.

## 22747 OPTIONS

22748 The *lp* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 22749 Section 12.2, Utility Syntax Guidelines.

22750 The following options shall be supported:

22751 **-c** Exit only after further access to any of the input files is no longer required. The  
 22752 application can then safely delete or modify the files without affecting the output  
 22753 operation. Normally, files are not copied, but are linked whenever possible. If the  
 22754 *-c* option is not given, then the user should be careful not to remove any of the  
 22755 files before the request has been printed in its entirety. It should also be noted that  
 22756 in the absence of the *-c* option, any changes made to the named files after the  
 22757 request is made but before it is printed are reflected in the printed output. On some  
 22758 systems, *-c* may be on by default.

22759 MAN **-d *dest*** Specify a string that names the destination (*dest*). If *dest* is a printer, the request  
 22760 shall be printed only on that specific printer. If *dest* is a class of printers, the request  
 22761 shall be printed on the first available printer that is a member of the class. Under  
 22762 certain conditions (printer unavailability, file space limitation, and so on), requests  
 22763 for specific destinations need not be accepted. Destination names vary between  
 22764 systems.

22765 If *-d* is not specified, and neither the *LPDEST* nor *PRINTER* environment variable  
 22766 is set, an unspecified destination is used. The *-d dest* option shall take precedence  
 22767 over *LPDEST*, which in turn shall take precedence over *PRINTER*. Results are  
 22768 undefined when *dest* contains a value that is not a valid destination name.

22769 MAN UN **-m** Send mail (see *mailx* on page 619) after the files have been printed. By default, no  
 22770 mail is sent upon normal completion of the print request.

22771 **-n *copies*** Write *copies* number of copies of the files, where *copies* is a positive decimal integer.  
 22772 The methods for producing multiple copies and for arranging the multiple copies  
 22773 when multiple *file* operands are used are unspecified, except that each file shall be  
 22774 output as an integral whole, not interleaved with portions of other files.

22775 MAN UN **-o option** Specify printer-dependent or class-dependent *options*. Several such *options* may be collected by specifying the **-o** option more than once.

22777 MAN PI **-s** Suppress messages from *lp* such as “request id is ...”.

22778 MAN UN **-t title** Write *title* on the banner page of the output.

22779 MAN UN **-w** Write a message on the user’s terminal after the files have been printed. If the user is not logged in, then mail shall be sent instead.

## 22781 OPERANDS

22782 The following operand shall be supported:

22783 *file* A path name of a file to be output. If no *file* operands are specified, or if a *file* operand is ‘-’, the standard input shall be used. If a *file* operand is used, but the  
22784  
22785 **-c** option is not specified, the process performing the writing to the output device  
22786 may have user and group permissions that differ from that of the process invoking  
22787 *lp*.

## 22788 STDIN

22789 The standard input is used only if no *file* operands are specified, or if a *file* operand is ‘-’. See  
22790 the INPUT FILES section.

## 22791 INPUT FILES

22792 The input files shall be text files.

## 22793 ENVIRONMENT VARIABLES

22794 The following environment variables shall affect the execution of *lp*:

22795 *LANG* Provide a default value for the internationalization variables that are unset or null.  
22796 If *LANG* is unset or null, the corresponding value from the implementation-  
22797 dependent default locale shall be used. If any of the internationalization variables  
22798 contains an invalid setting, the utility shall behave as if none of the variables had  
22799 been defined.

22800 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
22801 internationalization variables.

22802 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
22803 characters (for example, single-byte as opposed to multi-byte characters in  
22804 arguments and input files).

### 22805 LC\_MESSAGES

22806 Determine the locale that should be used to affect the format and contents of  
22807 diagnostic messages written to standard error and informative messages written to  
22808 standard output.

22809 MAN *LC\_TIME* Determine the format and contents of date and time strings displayed in the *lp*  
22810 banner page, if any.

22811 *LPDEST* Determine the destination. If the *LPDEST* environment variable is not set, the  
22812 *PRINTER* environment variable shall be used. The **-d dest** option takes precedence  
22813 over *LPDEST*. Results are undefined when **-d** is not specified and *LPDEST*  
22814 contains a value that is not a valid destination name.

22815 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

22816 *PRINTER* Determine the output device or destination. If the *LPDEST* and *PRINTER*  
22817 environment variables are not set, an unspecified output device is used. The **-d**  
22818 *dest* option and the *LPDEST* environment variable shall take precedence over

22819 *PRINTER*. Results are undefined when **-d** is not specified, *LPDEST* is unset, and  
 22820 *PRINTER* contains a value that is not a valid device or destination name.

## 22821 ASYNCHRONOUS EVENTS

22822 Default.

## 22823 STDOUT

22824 MAN The *lp* utility shall write a *request ID* to the standard output, unless **-s** is specified. The format of  
 22825 the message is unspecified. The request ID can be used on systems supporting the historical  
 22826 *cancel* and *lpstat* utilities.

## 22827 STDERR

22828 Used only for diagnostic messages.

## 22829 OUTPUT FILES

22830 None.

## 22831 EXTENDED DESCRIPTION

22832 None.

## 22833 EXIT STATUS

22834 The following exit values shall be returned:

- 22835 0 All input files were processed successfully.
- 22836 >0 No output device was available, or an error occurred.

## 22837 CONSEQUENCES OF ERRORS

22838 Default.

## 22839 APPLICATION USAGE

22840 The *pr* and *fold* utilities can be used to achieve reasonable formatting for the implementation's  
 22841 default page size.

22842 A portable application can use one of the *file* operands only with the **-c** option or if the file is  
 22843 publicly readable and guaranteed to be available at the time of printing. This is because the  
 22844 standard gives the implementation the freedom to queue up the request for printing at some  
 22845 later time by a different process that might not be able to access the file.

## 22846 EXAMPLES

- 22847 1. To print file *file*:  
 22848 `lp -c file`
- 22849 2. To print multiple files with headers:  
 22850 `pr file1 file2 | lp`

## 22851 RATIONALE

22852 The *lp* utility was designed to be a basic version of a utility that is already available in many  
 22853 historical implementations. The standard developers considered that it should be implementable  
 22854 simply as:

22855 `cat "$@" > /dev/lp`

22856 after appropriate processing of options, if that is how the implementation chose to do it and if  
 22857 exclusive access could be granted (so that two users did not write to the device simultaneously).  
 22858 Although in the future the standard developers may add other options to this utility, it should  
 22859 always be able to execute with no options or operands and send the standard input to an  
 22860 unspecified output device.

22861 This volume of IEEE Std. 1003.1-200x makes no representations concerning the format of the  
 22862 printed output, except that it must be “human-readable” and “non-volatile”. Thus, writing by  
 22863 default to a disk or tape drive or a display terminal would not qualify. (Such destinations are not  
 22864 prohibited when `-d dest`, `LPDEST`, or `PRINTER` are used, however.)

22865 This volume of IEEE Std. 1003.1-200x is worded such that a “print job” consisting of multiple  
 22866 input files, possibly in multiple copies, is guaranteed to print so that any one file is not  
 22867 intermixed with another, but there is no statement that all the files or copies have to print out  
 22868 together.

22869 The `-c` option may imply a spooling operation, but this is not required. The utility can be  
 22870 implemented to wait until the printer is ready and then wait until it is finished. Because of that,  
 22871 there is no attempt to define a queuing mechanism (priorities, classes of output, and so on).

22872 On some historical systems, the request ID reported on the `STDOUT` can be used to later cancel  
 22873 or find the status of a request using utilities not defined in this volume of IEEE Std. 1003.1-200x.

22874 Although the historical System V *lp* and BSD *lpr* utilities have provided similar functionality,  
 22875 they used different names for the environment variable specifying the destination printer. Since  
 22876 the name of the utility here is *lp*, `LPDEST` (used by the System V *lp* utility) was given precedence  
 22877 over `PRINTER` (used by the BSD *lpr* utility). Since environments of users frequently contain one  
 22878 or the other environment variable, the *lp* utility is required to recognize both. If this was not  
 22879 done, many applications would send output to unexpected output devices when users moved  
 22880 from system to system.

22881 Some have commented that *lp* has far too little functionality to make it worthwhile. Requests  
 22882 have proposed additional options or operands or both that added functionality. The requests  
 22883 included:

- 22884 • Wording *requiring* the output to be “hardcopy”
- 22885 • A requirement for multiple printers
- 22886 • Options for supporting various page-description languages

22887 Given that a compliant system is not required to even have a printer, placing further restrictions  
 22888 upon the behavior of the printer is not useful. Since hardcopy format is so application-  
 22889 dependent, it is difficult, if not impossible, to select a reasonable subset of functionality that  
 22890 should be required on all compliant systems.

22891 The term “unspecified” is used in this section in lieu of “implementation-dependent” as most  
 22892 known implementations would not be able to make definitive statements in their conformance  
 22893 documents: the existence and usage of printers is very dependent on how the system  
 22894 administrator configures each individual system.

22895 Since the default destination, device type, queuing mechanisms, and acceptable forms of input  
 22896 are all unspecified, usage guidelines for what a portable application can do are as follows:

- 22897 • Use the command in a pipeline, or with `-c`, so that there are no permission problems and the  
 22898 files can be safely deleted or modified.
- 22899 • Limit output to text files of reasonable line lengths and printable characters and include no  
 22900 device-specific formatting information, such as a page description language. The meaning of  
 22901 “reasonable” in this context can only be answered as a quality-of-implementation issue, but  
 22902 it should be apparent from historical usage patterns in the industry and the locale. The *pr* and  
 22903 *fold* utilities can be used to achieve reasonable formatting for the default page size of the  
 22904 implementation.

22905 Alternatively, the application can arrange its installation in such a way that it requires the  
 22906 system administrator or operator to provide the appropriate information on *lp* options and  
 22907 environment variable values.

22908 At a minimum, having this utility in this volume of IEEE Std. 1003.1-200x tells the industry that  
 22909 portable applications require a means to print output and provides at least a command name  
 22910 and *LPDEST* routing mechanism that can be used for discussions between vendors, application  
 22911 writers, and users. The use of “should” in the DESCRIPTION of *lp* clearly shows the intent of  
 22912 the standard developers, even if they cannot mandate that all systems (such as laptops) have  
 22913 printers.

22914 This volume of IEEE Std. 1003.1-200x does not specify what the ownership of the process  
 22915 performing the writing to the output device may be. If *-c* is not used, it is unspecified whether  
 22916 the process performing the writing to the output device has permission to read *file* if there are  
 22917 any restrictions in place on who may read *file* until after it is printed. Also, if *-c* is not used, the  
 22918 results of deleting *file* before it is printed are unspecified.

#### 22919 **FUTURE DIRECTIONS**

22920 None.

#### 22921 **SEE ALSO**

22922 *mailx*

#### 22923 **CHANGE HISTORY**

22924 First released in Issue 2.

#### 22925 **Issue 4**

22926 Aligned with the ISO/IEC 9945-2: 1993 standard.

#### 22927 **Issue 6**

22928 The following new requirements on POSIX implementations derive from alignment with the  
 22929 Single UNIX Specification:

- 22930 • In the DESCRIPTION, the requirement to associate a unique request ID, and the normal
- 22931 generation of a banner page is added.
- 22932 • In the OPTIONS section:
  - 22933 — The *-d dest* description is expanded, but references to *lpstat* are removed.
  - 22934 — The *-m*, *-o*, *-s*, *-t*, and *-w* options are added.
- 22935 • In the ENVIRONMENT VARIABLES section, *LC\_TIME* may now affect the execution.
- 22936 • The STDOUT section is added.

22937 The normative text is reworded to avoid use of the term “must” for application requirements.

## 22938 NAME

22939 ls — list directory contents

## 22940 SYNOPSIS

22941 xSI ls [-CFRacdilqrtul][-H | -L ][-fgmnopsx][file...]

## 22942 DESCRIPTION

22943 For each operand that names a file of a type other than directory or symbolic link to a directory,  
 22944 *ls* shall write the name of the file as well as any requested, associated information. For each  
 22945 operand that names a file of type directory, *ls* shall write the names of files contained within the  
 22946 directory as well as any requested, associated information. If one of the **-d**, **-F**, or **-l** options are  
 22947 specified, and one of the **-H** or **-L** options are not specified, for each operand that names a file of  
 22948 type symbolic link to a directory, *ls* shall write the name of the file as well as any requested,  
 22949 associated information. If none of the **-d**, **-F**, or **-l** options are specified, or the **-H** or **-L** options  
 22950 are specified, for each operand that names a file of type symbolic link to a directory, *ls* shall write  
 22951 the names of files contained within the directory as well as any requested, associated  
 22952 information.

22953 If no operands are specified, *ls* shall write the contents of the current directory. If more than one  
 22954 operand is specified, *ls* shall write non-directory operands first; it shall sort directory and non-  
 22955 directory operands separately according to the collating sequence in the current locale.

22956 The *ls* utility shall detect infinite loops; that is, entering a previously visited directory that is an  
 22957 ancestor of the last file encountered. When it detects an infinite loop, *ls* shall write a diagnostic  
 22958 message to standard error and shall either recover its position in the hierarchy or terminate.

## 22959 OPTIONS

22960 The *ls* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 22961 Section 12.2, Utility Syntax Guidelines.

22962 The following options shall be supported:

22963 **-C** Write multi-text-column output with entries sorted down the columns, according  
 22964 to the collating sequence. The number of text columns and the column separator  
 22965 characters are unspecified, but should be adapted to the nature of the output  
 22966 device.

22967 **-F** Do not follow symbolic links named as operands unless the **-H** or options are  
 22968 specified. Write a slash ( '/' ) immediately after each path name that is a directory,  
 22969 an asterisk ( '\*' ) after each that is executable, a vertical bar ( '|' ) after each that is  
 22970 a FIFO, and an at sign ( '@' ) after each that is a symbolic link. For other file types,  
 22971 other symbols may be written.

22972 **-H** If a symbolic link referencing a file of type directory is specified on the command  
 22973 line, *ls* shall evaluate the file information and file type to be those of the file  
 22974 referenced by the link, and not the link itself; however, *ls* shall write the name of  
 22975 the link itself and not the file referenced by the link.

22976 **-L** Evaluate the file information and file type for all symbolic links (whether named  
 22977 on the command line or encountered in a file hierarchy) to be those of the file  
 22978 referenced by the link, and not the link itself; however, *ls* shall write the name of  
 22979 the link itself and not the file referenced by the link. When **-L** is used with **-l**, write  
 22980 the contents of symbolic links in the long format (see the STDOUT section).

22981 **-R** Recursively list subdirectories encountered.

22982 **-a** Write out all directory entries, including those whose names begin with a period  
 22983 ( '.' ). Entries beginning with a period shall not be written out unless explicitly



|           |           |                                                                                                                                                                                                                                                                                                                                                                               |
|-----------|-----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 22984     |           | referenced, the <b>-a</b> option is supplied, or an implementation-dependent condition shall cause them to be written.                                                                                                                                                                                                                                                        |
| 22985     |           |                                                                                                                                                                                                                                                                                                                                                                               |
| 22986     | <b>-c</b> | Use time of last modification of the file status information (see < <b>sys/stat.h</b> > in the System Interfaces volume of IEEE Std. 1003.1-200x) instead of last modification of the file itself for sorting ( <b>-t</b> ) or writing ( <b>-l</b> ).                                                                                                                         |
| 22987     |           |                                                                                                                                                                                                                                                                                                                                                                               |
| 22988     |           |                                                                                                                                                                                                                                                                                                                                                                               |
| 22989     | <b>-d</b> | Do not follow symbolic links named as operands unless the <b>-H</b> or <b>-L</b> options are specified. Do not treat directories differently than other types of files. The use of <b>-d</b> with <b>-R</b> produces unspecified results.                                                                                                                                     |
| 22990     |           |                                                                                                                                                                                                                                                                                                                                                                               |
| 22991     |           |                                                                                                                                                                                                                                                                                                                                                                               |
| 22992 XSI | <b>-f</b> | Force each argument to be interpreted as a directory and list the name found in each slot. This option shall turn off <b>-l</b> , <b>-t</b> , <b>-s</b> , and <b>-r</b> , and shall turn on <b>-a</b> ; the order is the order in which entries appear in the directory.                                                                                                      |
| 22993     |           |                                                                                                                                                                                                                                                                                                                                                                               |
| 22994     |           |                                                                                                                                                                                                                                                                                                                                                                               |
| 22995 XSI | <b>-g</b> | The same as <b>-l</b> , except that the owner shall not be written.                                                                                                                                                                                                                                                                                                           |
| 22996     | <b>-i</b> | For each file, write the file's file serial number (see <i>stat()</i> in the System Interfaces volume of IEEE Std. 1003.1-200x).                                                                                                                                                                                                                                              |
| 22997     |           |                                                                                                                                                                                                                                                                                                                                                                               |
| 22998     | <b>-l</b> | (The letter ell.) Do not follow symbolic links named as operands unless the <b>-H</b> or <b>-L</b> options are specified. Write out in long format (see the STDOUT section). When <b>-l</b> (ell) is specified, <b>-1</b> (one) shall be assumed.                                                                                                                             |
| 22999     |           |                                                                                                                                                                                                                                                                                                                                                                               |
| 23000     |           |                                                                                                                                                                                                                                                                                                                                                                               |
| 23001 XSI | <b>-m</b> | Stream output format; list files across the page, separated by commas.                                                                                                                                                                                                                                                                                                        |
| 23002 XSI | <b>-n</b> | The same as <b>-l</b> , except that the owner's UID and GID numbers are written, rather than the associated character strings.                                                                                                                                                                                                                                                |
| 23003     |           |                                                                                                                                                                                                                                                                                                                                                                               |
| 23004 XSI | <b>-o</b> | The same as <b>-l</b> , except that the group is not written.                                                                                                                                                                                                                                                                                                                 |
| 23005 XSI | <b>-p</b> | Write a slash ( <b>/</b> ) after each file name if that file is a directory.                                                                                                                                                                                                                                                                                                  |
| 23006     | <b>-q</b> | Force each instance of non-printable file name characters and <tab> characters to be written as the question-mark ( <b>?</b> ) character. Implementations may provide this option by default if the output is to a terminal device.                                                                                                                                           |
| 23007     |           |                                                                                                                                                                                                                                                                                                                                                                               |
| 23008     |           |                                                                                                                                                                                                                                                                                                                                                                               |
| 23009     | <b>-r</b> | Reverse the order of the sort to get reverse collating sequence or oldest first.                                                                                                                                                                                                                                                                                              |
| 23010 XSI | <b>-s</b> | Indicate the total number of file system blocks consumed by each file displayed. The block size is implementation-dependent.                                                                                                                                                                                                                                                  |
| 23011     |           |                                                                                                                                                                                                                                                                                                                                                                               |
| 23012     | <b>-t</b> | Sort by time modified (most recently modified first) before sorting the operands by the collating sequence.                                                                                                                                                                                                                                                                   |
| 23013     |           |                                                                                                                                                                                                                                                                                                                                                                               |
| 23014     | <b>-u</b> | Use time of last access (see < <b>sys/stat.h</b> > in the System Interfaces volume of IEEE Std. 1003.1-200x) instead of last modification of the file for sorting ( <b>-t</b> ) or writing ( <b>-l</b> ).                                                                                                                                                                     |
| 23015     |           |                                                                                                                                                                                                                                                                                                                                                                               |
| 23016     |           |                                                                                                                                                                                                                                                                                                                                                                               |
| 23017 XSI | <b>-x</b> | The same as <b>-C</b> , except that the multi-text-column output is produced with entries sorted across, rather than down, the columns.                                                                                                                                                                                                                                       |
| 23018     |           |                                                                                                                                                                                                                                                                                                                                                                               |
| 23019     | <b>-q</b> | (The numeric digit one.) Force output to be one entry per line.                                                                                                                                                                                                                                                                                                               |
| 23020     |           | Specifying more than one of the options in the following mutually exclusive pairs shall not be considered an error: <b>-C</b> and <b>-l</b> (ell), <b>-m</b> and <b>-l</b> (ell), <b>-x</b> and <b>-l</b> (ell), <b>-C</b> and <b>-1</b> (one), <b>-H</b> and <b>-L</b> , <b>-c</b> and <b>-u</b> . The last option specified in each pair shall determine the output format. |
| 23021 XSI |           |                                                                                                                                                                                                                                                                                                                                                                               |
| 23022     |           |                                                                                                                                                                                                                                                                                                                                                                               |

23023 **OPERANDS**

23024 The following operand shall be supported:

23025 *file* A path name of a file to be written. If the file specified is not found, a diagnostic  
 23026 message shall be output on standard error.

23027 **STDIN**

23028 Not used.

23029 **INPUT FILES**

23030 None.

23031 **ENVIRONMENT VARIABLES**23032 The following environment variables shall affect the execution of *ls*:

23033 *COLUMNS* Determine the user's preferred column position width for writing multiple text-  
 23034 column output. If this variable contains a string representing a decimal integer, the  
 23035 *ls* utility shall calculate how many path name text columns to write (see **-C**) based  
 23036 on the width provided. If *COLUMNS* is not set or invalid, an implementation-  
 23037 dependent number of column positions shall be assumed, based on the  
 23038 implementation's knowledge of the output device. The column width chosen to  
 23039 write the names of files in any given directory shall be constant. File names shall  
 23040 not be truncated to fit into the multiple text-column output.

23041 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 23042 If *LANG* is unset or null, the corresponding value from the implementation-  
 23043 dependent default locale shall be used. If any of the internationalization variables  
 23044 contains an invalid setting, the utility shall behave as if none of the variables had  
 23045 been defined.

23046 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 23047 internationalization variables.

23048 *LC\_COLLATE*

23049 Determine the locale for character collation information in determining the path  
 23050 name collation sequence.

23051 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 23052 characters (for example, single-byte as opposed to multi-byte characters in  
 23053 arguments) and which characters are defined as printable (character class **print**).

23054 *LC\_MESSAGES*

23055 Determine the locale that should be used to affect the format and contents of  
 23056 diagnostic messages written to standard error.

23057 *LC\_TIME* Determine the format and contents for date and time strings written by *ls*.

23058 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

23059 *TZ* Determine the timezone for date and time strings written by *ls*.

23060 **ASYNCHRONOUS EVENTS**

23061 Default.

23062 **STDOUT**

23063 The default format shall be to list one entry per line to standard output; the exceptions are to  
 23064 XSI terminals or when one of the **-C**, **-m**, or **-x** options is specified. If the output is to a terminal, the  
 23065 format is implementation-dependent.

|       |     |                                                                                                                                  |
|-------|-----|----------------------------------------------------------------------------------------------------------------------------------|
| 23066 | XSI | When <b>-m</b> is specified, the format used shall be:                                                                           |
| 23067 |     | "%s, %s, ...\\n", <filename1>, <filename2>                                                                                       |
| 23068 |     | where the largest number of file names shall be written without exceeding the length of the line.                                |
| 23069 |     | If the <b>-i</b> option is specified, the file's file serial number (see <sys/stat.h> in the System Interfaces                   |
| 23070 |     | volume of IEEE Std. 1003.1-200x) shall be written in the following format before any other                                       |
| 23071 |     | output for the corresponding entry:                                                                                              |
| 23072 |     | %u ", <file serial number>                                                                                                       |
| 23073 |     | If the <b>-l</b> option is specified without <b>-L</b> , the following information shall be written:                             |
| 23074 |     | "%s %u %s %s %u %s %s\\n", <file mode>, <number of links>,                                                                       |
| 23075 |     | <owner name>, <group name>, <number of bytes in the file>,                                                                       |
| 23076 |     | <date and time>, <pathname>                                                                                                      |
| 23077 |     | If the file is a symbolic link, this information shall be about the link itself and the <pathname>                               |
| 23078 |     | field shall be of the form:                                                                                                      |
| 23079 |     | "%s -> %s", <pathname of link>, <contents of link>                                                                               |
| 23080 |     | If both <b>-l</b> and <b>-L</b> are specified, the following information shall be written:                                       |
| 23081 |     | "%s %u %s %s %u %s %s0, <file mode>, <number of links>,                                                                          |
| 23082 |     | <owner name>, <group name>, <number of bytes in the file>,                                                                       |
| 23083 |     | <date and time>, <pathname of link>                                                                                              |
| 23084 |     | where all fields except <pathname of link> shall be for the file resolved from the symbolic link.                                |
| 23085 | XSI | The <b>-g</b> , <b>-n</b> , and <b>-o</b> options use the same format as <b>-l</b> , but with omitted items and their            |
| 23086 |     | associated <blank> characters. See the OPTIONS section.                                                                          |
| 23087 | XSI | In both the preceding <b>-l</b> forms, If <owner name> or <group name> cannot be determined, or if <b>-n</b>                     |
| 23088 |     | is given, they shall be replaced with their associated numeric values using the format %u.                                       |
| 23089 |     | The <date and time> field shall contain the appropriate date and timestamp of when the file was                                  |
| 23090 |     | last modified. In the POSIX locale, the field shall be the equivalent of the output of the following                             |
| 23091 |     | date command:                                                                                                                    |
| 23092 |     | date "+%b %e %H:%M"                                                                                                              |
| 23093 |     | if the file has been modified in the last six months, or:                                                                        |
| 23094 |     | date "+%b %e %Y"                                                                                                                 |
| 23095 |     | (where two <space> characters are used between %e and %Y) if the file has not been modified in                                   |
| 23096 |     | the last six months or if the modification date is in the future, except that, in both cases, the final                          |
| 23097 |     | <newline> character produced by date shall not be included and the output shall be as if the date                                |
| 23098 |     | command were executed at the time of the last modification date of the file rather than the                                      |
| 23099 |     | current time. When the LC_TIME locale category is not set to the POSIX locale, a different format                                |
| 23100 |     | and order of presentation of this field may be used.                                                                             |
| 23101 |     | If the file is a character special or block special file, the size of the file may be replaced with                              |
| 23102 |     | implementation-dependent information associated with the device in question.                                                     |
| 23103 |     | If the path name was specified as a file operand, it shall be written as specified.                                              |
| 23104 | XSI | The file mode written under the <b>-l</b> , <b>-g</b> , <b>-n</b> , and <b>-o</b> options shall consist of the following format: |
| 23105 |     | "%c%s%s%c", <entry type>, <owner permissions>,                                                                                   |
| 23106 |     | <group permissions>, <other permissions>,                                                                                        |

23107           *<optional alternate access method flag>*

23108       The *<optional alternate access method flag>* shall be a single *<space>* character if there is no  
 23109       alternate or additional access control method associated with the file; otherwise, a printable  
 23110       character shall be used.

23111       The *<entry type>* character shall describe the type of file, as follows:

23112       d       Directory.

23113       b       Block special file.

23114       c       Character special file.

23115       l (ell)   Symbolic link.

23116       p       FIFO.

23117       –       Regular file.

23118       Implementations may add other characters to this list to represent other implementation-  
 23119       dependent file types.

23120       The next three fields shall be three characters each:

23121       *<owner permissions>*  
 23122       Permissions for the file owner class (see the System Interface Definitions volume of  
 23123       IEEE Std. 1003.1-200x, Section 4.1, File Access Permissions).

23124       *<group permissions>*  
 23125       Permissions for the file group class.

23126       *<other permissions>*  
 23127       Permissions for the file other class.

23128       Each field shall have three character positions:

23129       1. If 'r', the file is readable; if '–', the file is not readable.

23130       2. If 'w', the file is writable; if '–', the file is not writable.

23131       3. The first of the following that applies:

23132       s       If in *<owner permissions>*, the file is not executable and set-user-ID mode is set. If in  
 23133       *<group permissions>*, the file is not executable and set-group-ID mode is set.

23134       S       If in *<owner permissions>*, the file is executable and set-user-ID mode is set. If in  
 23135       *<group permissions>*, the file is executable and set-group-ID mode is set.

23136       x       The file is executable or the directory is searchable.

23137       –       None of the attributes of 'S', 's', or 'x' applies.

23138       Implementations may add other characters to this list for the third character position. Such  
 23139       additions shall, however, be written in lowercase if the file is executable or searchable, and  
 23140       in uppercase if it is not.

23141 xSI       If any of the **–l**, **–g**, **–n**, **–o**, or **–s** options is specified, each list of files within the directory shall be  
 23142       preceded by a status line indicating the number of file system blocks occupied by files in the  
 23143       directory in 512-byte units, rounded up to the next integral number of units, if necessary. In the  
 23144       POSIX locale, the format shall be:

23145       "total %u\n", *<number of units in the directory>*

23146 If more than one directory, or a combination of non-directory files and directories are written,  
 23147 either as a result of specifying multiple operands, or the **-R** option, each list of files within a  
 23148 directory shall be preceded by:

23149 "\n%s:\n", <directory name>

23150 If this string is the first thing to be written, the first <newline> character shall not be written.  
 23151 This output shall precede the number of units in the directory.

23152 XSI If the **-s** option is given, each file shall be written with the number of blocks used by the file.  
 23153 Along with **-C**, **-l**, **-m**, or **-x**, the number and a <space> character shall precede the file name;  
 23154 with **-g**, **-l**, **-n**, or **-o**, they shall precede each line describing a file.

#### 23155 **STDERR**

23156 Used only for diagnostic messages.

#### 23157 **OUTPUT FILES**

23158 None.

#### 23159 **EXTENDED DESCRIPTION**

23160 None.

#### 23161 **EXIT STATUS**

23162 The following exit values shall be returned:

23163 0 Successful completion.

23164 >0 An error occurred.

#### 23165 **CONSEQUENCES OF ERRORS**

23166 Default.

#### 23167 **APPLICATION USAGE**

23168 Many implementations use the equal sign ('=') and the at sign('@') to denote sockets bound to  
 23169 the file system and symbolic links, respectively, for the **-F** option. Similarly, many historical  
 23170 implementations use the 's' character and the 'l' character to denote sockets and symbolic  
 23171 links, respectively, as the entry type characters for the **-l** option.

23172 It is difficult for an application to use every part of the file modes field of *ls -l* in a portable  
 23173 manner. Certain file types and executable bits are not guaranteed to be exactly as shown, as  
 23174 implementations may have extensions. Applications can use this field to pass directly to a user  
 23175 printout or prompt, but actions based on its contents should generally be deferred, instead, to  
 23176 the *test* utility.

23177 The output of *ls* (with the **-l** and related options) contains information that logically could be  
 23178 used by utilities such as *chmod* and *touch* to restore files to a known state. However, this  
 23179 information is presented in a format that cannot be used directly by those utilities or be easily  
 23180 translated into a format that can be used. A character has been added to the end of the  
 23181 permissions string so that applications at least have an indication that they may be working in  
 23182 an area they do not understand instead of assuming that they can translate the permissions  
 23183 string into something that can be used. Future issues or related documents may define one or  
 23184 more specific characters to be used based on different standard additional or alternative access  
 23185 control mechanisms.

23186 As with many of the utilities that deal with file names, the output of *ls* for multiple files or in one  
 23187 of the long listing formats must be used carefully on systems where file names can contain  
 23188 embedded white space. Systems and system administrators should institute policies and user  
 23189 training to limit the use of such file names.

23190 The number of disk blocks occupied by the file that it reports varies depending on underlying  
 23191 file system type, block size units reported, and the method of calculating the number of blocks.  
 23192 On some file system types, the number is the actual number of blocks occupied by the file  
 23193 (counting indirect blocks and ignoring holes in the file); on others it is calculated based on the  
 23194 file size (usually making an allowance for indirect blocks, but ignoring holes).

#### 23195 EXAMPLES

23196 An example of a small directory tree being fully listed with *ls -laRF a* in the POSIX locale:

```
23197 total 11
23198 drwxr-xr-x 3 hlj prog 64 Jul 4 12:07 ./
23199 drwxrwxrwx 4 hlj prog 3264 Jul 4 12:09 ../
23200 drwxr-xr-x 2 hlj prog 48 Jul 4 12:07 b/
23201 -rwxr--r-- 1 hlj prog 572 Jul 4 12:07 foo*

23202 a/b:
23203 total 4
23204 drwxr-xr-x 2 hlj prog 48 Jul 4 12:07 ./
23205 drwxr-xr-x 3 hlj prog 64 Jul 4 12:07 ../
23206 -rw-r--r-- 1 hlj prog 700 Jul 4 12:07 bar
```

#### 23207 RATIONALE

23208 Some historical implementations of the *ls* utility show all entries in a directory except dot and  
 23209 dot-dot when a superuser invokes *ls* without specifying the *-a* option. When “normal” users  
 23210 invoke *ls* without specifying *-a*, they should not see information about any files with names  
 23211 beginning with period unless they were named as file operands.

23212 Implementations are expected to traverse arbitrary depths when processing the *-R* option. The  
 23213 only limitation on depth should be based on running out of physical storage for keeping track of  
 23214 untraversed directories.

23215 The *-1* (one) option is currently found in BSD and BSD-derived implementations only. It is  
 23216 required in this volume of IEEE Std. 1003.1-200x so that portable applications might ensure that  
 23217 output is one entry per line, even if the output is to a terminal.

23218 Generally, this volume of IEEE Std. 1003.1-200x is silent about what happens when options are  
 23219 given multiple times. In the cases of *-C*, *-l*, and *-1*, however, it does specify the results of these  
 23220 overlapping options. Since *ls* is one of the most aliased commands, it is important that the  
 23221 implementation perform intuitively. For example, if the alias were:

```
23222 alias ls="ls -C"
```

23223 and the user typed *ls -1*, single-text-column output should result, not an error.

23224 The BSD *ls* provides a *-A* option (like *-a*, but dot and dot-dot are not written out). The small  
 23225 difference from *-a* did not seem important enough to require both.

23226 Implementations are allowed to make *-q* the default for terminals to prevent trojan horse  
 23227 attacks on terminals with special escape sequences. This is not required because:

- 23228 • Some control characters may be useful on some terminals; for example, a system might write  
 23229 them as "\001" or "^A".

- 23230 • Special behavior for terminals is not relevant to application portability.

23231 An early proposal specified that the optional alternate access method flag had to be '+' if there  
 23232 was an alternate access method used on the file or <space> if there was not. This was changed to  
 23233 be <space> if there is not and a single printable character if there is. This was done for three  
 23234 reasons:

- 23235 1. There are historical implementations using characters other than '+'.
- 23236 2. There are implementations that vary this character used in that position to distinguish
- 23237 between various alternate access methods in use.
- 23238 3. The standard developers did not want to preclude futures specifications that might need a
- 23239 way to specify more than one alternate access method.
- 23240 Nonetheless, implementations providing a single alternate access method are encouraged to use
- 23241 '+'.
- 23242 In an early proposal, the units used to specify the number of blocks occupied by files in a
- 23243 directory in an *ls -l* listing was implementation-dependent. This was because BSD systems have
- 23244 historically used 1024-byte units and System V systems have historically used 512-byte units. It
- 23245 was pointed out by BSD developers that their system has used 512-byte units in some places and
- 23246 1024-byte units in other places. (System V has consistently used 512.) Therefore, this volume of
- 23247 IEEE Std. 1003.1-200x usually specifies 512. Future releases of BSD are expected to consistently
- 23248 provide 512 bytes as a default with a way of specifying 1024-byte units where appropriate.
- 23249 The *<date and time>* field in the *-l* format is specified only for the POSIX locale. As noted, the
- 23250 format can be different in other locales. No mechanism for defining this is present in this volume
- 23251 of IEEE Std. 1003.1-200x, as the appropriate vehicle is a messaging system; that is, the format
- 23252 should be specified as a "message".
- 23253 **FUTURE DIRECTIONS**
- 23254 The *-s* uses implementation-dependent units and cannot be used portably; it may be withdrawn
- 23255 in a future issue.
- 23256 **SEE ALSO**
- 23257 *chmod*, *find*, the System Interfaces volume of IEEE Std. 1003.1-200x, *<sys/stat.h>*
- 23258 **CHANGE HISTORY**
- 23259 First released in Issue 2.
- 23260 **Issue 4**
- 23261 Aligned with the ISO/IEC 9945-2:1993 standard.
- 23262 **Issue 5**
- 23263 Second FUTURE DIRECTION added.
- 23264 **Issue 6**
- 23265 The following new requirements on POSIX implementations derive from alignment with the
- 23266 Single UNIX Specification:
- 23267 • In the *-F* option, other symbols are allowed for other file types.
- 23268 Treatment of symbolic links is added, as defined in the IEEE P1003.2b draft standard.

23269 **NAME**23270 m4 — macro processor (**DEVELOPMENT**)23271 **SYNOPSIS**23272 xSI m4 [-s][-D *name*[=*val*]]...[-U *name*]... *file*...

23273

23274 **DESCRIPTION**

23275 The *m4* utility is a macro processor that shall read one or more text files, process them according  
 23276 to their included macro statements, and write the results to standard output.

23277 **OPTIONS**

23278 The *m4* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 23279 Section 12.2, Utility Syntax Guidelines, except that the order of the **-D** and **-U** options shall be  
 23280 significant.

23281 The following options shall be supported:

23282 **-s** Enable line synchronization output for the *c89* preprocessor phase (that is, **#line**  
 23283 directives).

23284 **-D *name*[=*val*]**23285 Define *name* to *val* or to null if *=val* is omitted.23286 **-U *name*** Undefine *name*.23287 **OPERANDS**

23288 The following operand shall be supported:

23289 *file* A path name of a text file to be processed. If no *file* is given, or if it is '-', the  
 23290 standard input shall be read.

23291 **STDIN**23292 The standard input shall be a text file that is used if no *file* operand is given, or if it is '-'.23293 **INPUT FILES**23294 The input file named by the *file* operand shall be a text file.23295 **ENVIRONMENT VARIABLES**23296 The following environment variables shall affect the execution of *m4*:

23297 **LANG** Provide a default value for the internationalization variables that are unset or null.  
 23298 If *LANG* is unset or null, the corresponding value from the implementation-  
 23299 dependent default locale shall be used. If any of the internationalization variables  
 23300 contains an invalid setting, the utility shall behave as if none of the variables had  
 23301 been defined.

23302 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 23303 internationalization variables.

23304 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 23305 characters (for example, single-byte as opposed to multi-byte characters in  
 23306 arguments and input files).

23307 **LC\_MESSAGES**

23308 Determine the locale that should be used to affect the format and contents of  
 23309 diagnostic messages written to standard error.

23310 **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.



23311 **ASYNCHRONOUS EVENTS**

23312 Default.

23313 **STDOUT**

23314 The standard output shall be the same as the input files, after being processed for macro  
 23315 expansion.

23316 **STDERR**

23317 Used to display strings with the **errprint** macro, macro tracing enabled by the **traceon** macro, the  
 23318 defined text for macros written by the **dumpdef** macro, or for diagnostic messages.

23319 **OUTPUT FILES**

23320 None.

23321 **EXTENDED DESCRIPTION**

23322 The *m4* utility shall compare each token from the input against the set of built-in and user-  
 23323 defined macros. If the token matches the name of a macro, then the token shall be replaced by  
 23324 the macros defining text, if any, and rescanned for matching macro names. Once no portion of  
 23325 the token matches the name of a macro, it shall be written to standard output. Macros may have  
 23326 arguments, in which case the arguments shall be substituted into the defining text before it is  
 23327 rescanned.

23328 Macro calls have the form:

23329 *name(arg1, arg2, ..., argn)*

23330 Macro names shall consist of letters, digits, and underscores, where the first character is not a  
 23331 digit. Tokens not of this form shall not be treated as macro names.

23332 The application shall ensure that the left parenthesis immediately follows the name of the  
 23333 macro. If a token matching the name of a macro is not followed by a left parenthesis, it is  
 23334 handled as a use of that macro without arguments.

23335 If a macro name is followed by a left parenthesis, its arguments are the comma-separated tokens  
 23336 between the left parenthesis and the matching right parenthesis. Unquoted <blank> and  
 23337 <newline> characters preceding each argument shall be ignored. All other characters, including  
 23338 trailing <blank> and <newline> characters, are retained. Commas enclosed between left and  
 23339 right parenthesis characters do not delimit arguments.

23340 Arguments are positionally defined and referenced. The string "\$1" in the defining text shall be  
 23341 replaced by the first argument. Systems shall support at least nine arguments; only the first nine  
 23342 can be referenced, using the strings "\$1" to "\$9", inclusive. The string "\$0" is replaced with  
 23343 the name of the macro. The string "\$#" is replaced by the number of arguments as a string. The  
 23344 string "\$\*" is replaced by a list of all of the arguments, separated by commas. The string "\$@"  
 23345 is replaced by a list of all of the arguments separated by commas, and each argument is quoted  
 23346 using the current left and right quoting strings.

23347 If fewer arguments are supplied than are in the macro definition, the omitted arguments are  
 23348 taken to be null. It is not an error if more arguments are supplied than are in the macro  
 23349 definition.

23350 No special meaning is given to any characters enclosed between matching left and right quoting  
 23351 strings, but the quoting strings are themselves discarded. By default, the left quoting string  
 23352 consists of a grave accent (``) and the right quoting string consists of an acute accent (') see  
 23353 also the **changequote** macro.

23354 Comments are written but not scanned for matching macro names; by default, the begin-  
 23355 comment string consists of the number sign character and the end-comment string consists of a  
 23356 <newline> character. See also the **changecom** and **dnl** macros.

23357 The *m4* utility makes available the following built-in macros. They can be redefined, but once  
 23358 this is done the original meaning is lost. Their values are null unless otherwise stated. In the  
 23359 descriptions below, the term *defining text* refers to the value of the macro: the second argument  
 23360 to the **define** macro, among other things.

23361 **changeocom** The **changeocom** macro sets the begin-comment and end-comment strings. With no  
 23362 arguments, the comment mechanism is disabled. With a single argument, that  
 23363 argument becomes the begin-comment string and the <newline> character  
 23364 becomes the end-comment string. With two arguments, the first argument  
 23365 becomes the begin-comment string and the second argument becomes the end-  
 23366 comment string. Systems support comment strings of at least five characters.

23367 **changequote** The **changequote** macro sets the begin-quote and end-quote strings. With no  
 23368 arguments, the quote strings are set to the default values (that is, ` `'). With a  
 23369 single argument, that argument becomes the begin-quote string and the <newline>  
 23370 character becomes the end-quote string. With two arguments, the first argument  
 23371 becomes the begin-quote string and the second argument becomes the end-quote  
 23372 string. Systems support quote strings of at least five characters.

23373 **decr** The defining text of the **decr** macro is its first argument decremented by 1. It is an  
 23374 error to specify an argument containing any non-numeric characters.

## 23375 *Notes to Reviewers*

23376 *This section with side shading will not appear in the final copy. - Ed.*

23377 Re D1, XCU, ERN 285: What base is this sort of arithmetic performed in: decimal,  
 23378 octal, or what? (Same for incr, eval, etc.) Is the output base retained?

23379 **define** The second argument is specified as the defining text of the macro whose name is  
 23380 the first argument.

23381 **defn** The defining text of the **defn** macro is the quoted definition (using the current  
 23382 quoting strings) of its arguments.

23383 **divert** The *m4* utility maintains ten temporary buffers, numbered 0 to 9, inclusive.

## 23384 *Notes to Reviewers*

23385 *This section with side shading will not appear in the final copy. - Ed.*

23386 Re D1, XCU, ERN 286: Buffer 0 seems strange: it's one of the 10 buffers, and thus  
 23387 should be a diversion buffer, but at 19704 it implies that it's the name of the main  
 23388 output. What is it (or are there really only 9 diversion buffers?) Also, see austin-  
 23389 group mail sequence #295.

23390 When the last of the input has been processed, any output that has been placed in  
 23391 these buffers is written to standard output in buffer-numerical order. The **divert**  
 23392 macro diverts future output to the buffer specified by its argument. Specifying no  
 23393 argument or an argument of 0 resumes the normal output process. Output  
 23394 diverted to a stream other than 0 to 9 is discarded. It is an error to specify an  
 23395 argument containing any non-numeric characters.

23396 **divnum** The defining text of the **divnum** macro is the number of the current output stream  
 23397 as a string.

23398 **dnl** The **dnl** macro shall cause *m4* to discard all input characters up to and including  
 23399 the next <newline> character.

|       |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 23400 | <b>dumpdef</b>            | The <b>dumpdef</b> macro writes the defined text to standard error for each of the macros specified as arguments, or, if no arguments are specified, for all macros.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 23401 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23402 | <b>errprint</b>           | The <b>errprint</b> macro writes its arguments to standard error.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 23403 | <b>eval</b>               | The <b>eval</b> macro evaluates its first argument as an arithmetic expression, using 32-bit signed integer arithmetic. All of the C-language operators are supported, except for "[ ]", "->", "++", "--", (type), unary '*', <b>sizeof</b> , ' ', ' . ', "?:", '&', and all assignment operators. It is an error to specify any of these operators. Precedence and associativity are as in C. Systems support octal and hexadecimal numbers as in C. The second argument, if specified, sets the radix for the result; the default is 10. The third argument, if specified, sets the minimum number of digits in the result. It is an error to specify the second or third argument containing any non-numeric characters. |
| 23404 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23405 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23406 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23407 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23408 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23409 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23410 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23411 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23412 | <b>ifdef</b>              | If the first argument to the <b>ifdef</b> macro is defined, the defining text is the second argument. Otherwise, the defining text is the third argument, if specified, or the null string, if not.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 23413 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23414 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23415 | <b>ifelse</b>             | If the first argument (or the defining text of the first argument if it is a macro name) to the <b>ifelse</b> macro is the same as the second argument (or the defining text of the second argument if it is a macro name), then the defining text is the third argument.                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 23416 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23417 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23418 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23419 | <b>Notes to Reviewers</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23420 |                           | <i>This section with side shading will not appear in the final copy. - Ed.</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 23421 |                           | D1, XCU, ERN 287 (as modified by email #297) suggests the following replacement text for <b>ifelse</b> : "This function takes 3n+0 or 3n+1 arguments. For each group of 3 arguments, if the first and second are the same, the result is the third of the group. If the strings are not equal, and no arguments remain, the defining text is null. If one argument remains, it becomes the defining text. If three or more arguments remain, the process is repeated with the new group of three arguments. If 3n+2 arguments are provided, the evaluation proceeds as above, but a warning is generated and the last argument ignored.                                                                                     |
| 23422 |                           | If there are more than four arguments, the initial comparison of the first and second arguments are repeated for each group of three arguments. If no match is found, the defining text is the argument following the last set of three compared; otherwise, it is null.                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 23423 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23424 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23425 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23426 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23427 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23428 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23429 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23430 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23431 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23432 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23433 | <b>include</b>            | The defining text for the <b>include</b> macro is the contents of the file named by the first argument. It is an error if the file cannot be read.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 23434 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23435 | <b>incr</b>               | The defining text of the <b>incr</b> macro is its first argument incremented by 1. It is an error to specify an argument containing any non-numeric characters.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23436 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23437 | <b>index</b>              | The defining text of the <b>index</b> macro is the first character position (as a string) in the first argument where a string matching the second argument begins (zero origin), or                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 23438 |                           | –1 if the second argument does not occur.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 23439 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23440 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23441 | <b>len</b>                | The defining text of the <b>len</b> macro is the length (as a string) of the first argument.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 23442 | <b>m4exit</b>             | Exit from the <i>m4</i> utility. If the first argument is specified, it is the exit code. The default is zero. It is an error to specify an argument containing any non-numeric characters.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 23443 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23444 |                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

|       |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------|----------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 23445 | <b>m4wrap</b>                                | The first argument is processed when EOF is reached. If the <b>m4wrap</b> macro is used multiple times, the arguments specified are processed in the order in which the <b>m4wrap</b> macros were processed.                                                                                                                                                                                                                                                                                                                                                        |
| 23446 |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23447 |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23448 | <b>maketemp</b>                              | The defining text is the first argument, with any trailing 'x' characters replaced with the current process ID as a string.                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 23449 |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23450 | <b>popdef</b>                                | The <b>popdef</b> macro deletes the current definition of its arguments, replacing it with the previous one. If there is no previous definition, the macro is undefined.                                                                                                                                                                                                                                                                                                                                                                                            |
| 23451 |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23452 | <b>pushdef</b>                               | The <b>pushdef</b> macro is identical to the <b>define</b> macro with the exception that it preserves any current definition for future retrieval using the <b>popdef</b> macro.                                                                                                                                                                                                                                                                                                                                                                                    |
| 23453 |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23454 | <b>shift</b>                                 | The defining text for the <b>shift</b> macro is all of its arguments except for the first one.                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 23455 | <b>sinclude</b>                              | The <b>sinclude</b> macro is identical to the <b>include</b> macro, except that it is not an error if the file is inaccessible.                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23456 |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23457 | <b>substr</b>                                | The defining text for the <b>substr</b> macro is the substring of the first argument beginning at the zero-offset character position specified by the second argument. The third argument, if specified, is the number of characters to select; if not specified, the characters from the starting point to the end of the first argument become the defining text. It is not an error to specify a starting point beyond the end of the first argument and the defining text is null. It is an error to specify an argument containing any non-numeric characters. |
| 23458 |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23459 |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23460 |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23461 |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23462 |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23463 |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23464 | <b>syscmd</b>                                | The <b>syscmd</b> macro interprets its first argument as a shell command line. The defining text is the string result of that command. No output redirection is performed by the <i>m4</i> utility. The exit status value from the command can be retrieved using the <b>sysval</b> macro.                                                                                                                                                                                                                                                                          |
| 23465 |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23466 |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23467 |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23468 | <b>sysval</b>                                | The defining text of the <b>sysval</b> macro is the exit value of the utility last invoked by the <b>syscmd</b> macro (as a string).                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 23469 |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23470 | <b>traceon</b>                               | The <b>traceon</b> macro enables tracing for the macros specified as arguments, or, if no arguments are specified, for all macros. The trace output is written to standard error in an unspecified format.                                                                                                                                                                                                                                                                                                                                                          |
| 23471 |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23472 |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23473 | <b>traceoff</b>                              | The <b>traceoff</b> macro disables tracing for the macros specified as arguments, or, if no arguments are specified, for all macros.                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 23474 |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23475 | <b>translit</b>                              | The defining text of the <b>translit</b> macro is the first argument with every character that occurs in the second argument replaced with the corresponding character from the third argument.                                                                                                                                                                                                                                                                                                                                                                     |
| 23476 |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23477 |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23478 | <b>undefine</b>                              | The <b>undefine</b> macro deletes all definitions (including those preserved using the <b>pushdef</b> macro) of the macros named by its arguments.                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 23479 |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23480 | <b>undivert</b>                              | The <b>undivert</b> macro shall cause immediate output of any text in temporary buffers named as arguments, or all temporary buffers if no arguments are specified. Buffers can be undiverted into other temporary buffers. Undiverting discards the contents of the temporary buffer. It is an error to specify an argument containing any non-numeric characters.                                                                                                                                                                                                 |
| 23481 |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23482 |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23483 |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23484 |                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23485 | <b>EXIT STATUS</b>                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23486 | The following exit values shall be returned: |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23487 | 0                                            | Successful completion.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

23488 >0 An error occurred

23489 If the **m4exit** macro is used, the exit value can be specified by the input file.

## 23490 CONSEQUENCES OF ERRORS

23491 Default.

## 23492 APPLICATION USAGE

23493 The **defn** macro is useful for renaming macros, especially built-ins.

## 23494 EXAMPLES

23495 An example of a single *m4* input file capable of generating two output files follows. The file  
23496 **file1.m4** could contain lines such as:

23497 `if(VER, 1, do_something)`

23498 `if(VER, 2, do_something)`

23499 The makefile for the program might include:

23500 `file1.1.c : file1.m4`

23501 `m4 -D VER=1 file1.m4 > file1.1.c`

23502 `...`

23503 `file1.2.c : file1.m4`

23504 `m4 -D VER=2 file1.m4 > file1.2.c`

23505 `...`

23506 The **-U** option can be used to undefine **VER**. If **file1.m4** contains:

23507 `if(VER, 1, do_something)`

23508 `if(VER, 2, do_something)`

23509 `ifndef(VER, do_something)`

23510 then the makefile would contain:

23511 `file1.0.c : file1.m4`

23512 `m4 -U VER file1.m4 > file1.0.c`

23513 `...`

23514 `file1.1.c : file1.m4`

23515 `m4 -D VER=1 file1.m4 > file1.1.c`

23516 `...`

23517 `file1.2.c : file1.m4`

23518 `m4 -D VER=2 file1.m4 > file1.2.c`

23519 `...`

## 23520 RATIONALE

23521 None.

## 23522 FUTURE DIRECTIONS

23523 None.

## 23524 SEE ALSO

23525 *c89*

## 23526 CHANGE HISTORY

23527 First released in Issue 2.

## 23528 Issue 4

23529 Format reorganized.

23530 Utility Syntax Guideline support mandated.

23531 Internationalized environment variable support mandated.

23532 **Issue 5**

23533 The phrase “the defined text for macros written by the **dumpdef** macro” is added to the  
23534 description of **STDERR**, and the description of **dumpdef** is updated to indicate that output is  
23535 written to standard error. The description of **eval** is updated to indicate that the list of excluded  
23536 C operators excludes unary **&** and **.**. In the description of **ifdef**, the phrase “and it is not  
23537 defined to be zero” is deleted.

23538 **Issue 6**

23539 In the EXTENDED DESCRIPTION, the **eval** text is updated to include a **&** character in the  
23540 excepted list. |

23541 The normative text is reworded to avoid use of the term “must” for application requirements. |

23542 **NAME**

23543           mailx — process messages

23544 **SYNOPSIS**23545           **Send Mode**23546           mailx [-s *subject*] *address...*23547           **Receive Mode**

23548           mailx -e

23549 MAN      mailx [-HiNn] [-F] [-u *user*]23550 MAN      mailx -f [-HiNn] [-F] [*file*]23551 **DESCRIPTION**

23552           The *mailx* utility provides a message sending and receiving facility. It has two major modes,  
 23553           selected by the options used: Send Mode and Receive Mode.

23554           On systems that do not support the User Portability Utilities option, an application using *mailx*  
 23555           shall have the ability to send messages in an unspecified manner (Send Mode). Unless the first  
 23556           character of one or more lines is tilde ('~'), all characters in the input message shall appear in  
 23557           the delivered message, but additional characters may be inserted in the message before it is  
 23558           retrieved.

23559           On systems supporting the User Portability Utilities option, mail-receiving capabilities and other  
 23560           interactive features, Receive Mode, described below, also shall be enabled.

23561           **Send Mode**

23562           Send Mode can be used by applications or users to send messages from the text in standard  
 23563           input.

23564           **Receive Mode**

23565           Receive Mode is more oriented to interactive users. Mail can be read and sent in this interactive  
 23566           mode.

23567           When reading mail, *mailx* provides commands to facilitate saving, deleting, and responding to  
 23568           messages. When sending mail, *mailx* allows editing, reviewing, and other modification of the  
 23569           message as it is entered.

23570           Incoming mail shall be stored in one or more unspecified locations for each user, collectively  
 23571           called the system *mailbox* for that user. When *mailx* is invoked in Receive Mode, the system  
 23572           mailbox shall be the default place to find new mail. As messages are read, they shall be marked  
 23573           to be moved to a secondary file for storage, unless specific action is taken. This secondary file is  
 23574           called the **mbox** and is normally located in the directory referred to by the *HOME* environment  
 23575           variable (see *MBOX* in the ENVIRONMENT VARIABLES section for a description of this file).  
 23576           Messages shall remain in this file until explicitly removed. When the -f option is used to read  
 23577           mail messages from secondary files, messages shall be retained in those files unless specifically  
 23578           removed. All three of these locations—system mailbox, **mbox**, and secondary file—are referred  
 23579           to in this section as simply “mailboxes”, unless more specific identification is required.

## 23580 OPTIONS

23581 The *mailx* utility shall conform to the System Interface Definitions volume of  
23582 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

23583 The following options shall be supported. (Only the **-s subject** option shall be required on all  
23584 systems. The other options are required only on systems supporting the User Portability Utilities  
23585 option.)

23586 **-e** Test for the presence of mail in the system mailbox. The *mailx* utility shall write  
23587 nothing and exit with a successful return code if there is mail to read.

23588 **-** Read messages from the file named by the *file* operand instead of the system  
23589 mailbox. (See also **folder**.) If no *file* operand is specified, read messages from the  
23590 **mbox** instead of the system mailbox.

23591 MAN **-F** Record the message in a file named after the first recipient. The name is the login-  
23592 name portion of the address found first on the **To:** line in the mail header.  
23593 Overrides the **record** variable, if set (see **Internal Variables in mailx** on page 626.)

23594 **-H** Write a header summary only.

23595 **-i** Ignore interrupts. (See also **ignore**).

23596 **-n** Do not initialize from the system default start-up file. See the EXTENDED  
23597 DESCRIPTION section.

23598 **-N** Do not write an initial header summary.

23599 **-s subject** Set the **Subject** header field to *subject*. All characters in the *subject* string shall  
23600 appear in the delivered message. The results are unspecified if *subject* is longer  
23601 than {LINE\_MAX} – 10 bytes or contains a <newline> character.

23602 **-u user** Read the system mailbox of the login name *user*. This shall only be successful if  
23603 the invoking user has the appropriate privileges to read the system mailbox of that  
23604 user.

## 23605 OPERANDS

23606 The following operands shall be supported:

23607 *address* Addressee of message. When **-n** is specified and no user start-up files are accessed  
23608 (see the EXTENDED DESCRIPTION section), the user or application shall ensure  
23609 this is an address to pass to the mail delivery system. Any system or user start-up  
23610 files may enable aliases (see **alias** under **Commands in mailx** on page 629) that  
23611 may modify the form of *address* before it is passed to the mail delivery system.

23612 *file* A path name of a file to be read instead of the system mailbox when **-f** is specified.  
23613 The meaning of the *file* option-argument shall be affected by the contents of the  
23614 **folder** internal variable; see **Internal Variables in mailx** on page 626.

## 23615 STDIN

23616 When *mailx* is invoked in Send Mode (the first synopsis line), standard input shall be the  
23617 message to be delivered to the specified addresses. When in Receive Mode, user commands are  
23618 accepted from *stdin*. If the User Portability Utilities option is not supported, standard input lines  
23619 beginning with a tilde ( '~ ') character produce unspecified results.

23620 If the User Portability Utilities option is supported, then in both Send and Receive Modes,  
23621 standard input lines beginning with the escape character (usually tilde ( '~ ')) affect processing  
23622 as described in **Command Escapes in mailx** on page 637.



## 23623 INPUT FILES

23624 When *mailx* is used as described by this volume of IEEE Std. 1003.1-200x, the *file* option-  
 23625 argument (see the *-f* option) and the **mbox** shall be text files containing mail messages,  
 23626 formatted as described in the OUTPUT FILES section. The nature of the system mailbox is  
 23627 unspecified; it need not be a file.

## 23628 ENVIRONMENT VARIABLES

23629 The following environment variables shall affect the execution of *mailx*:

23630 **DEAD** Determine the path name of the file in which to save partial messages in case of  
 23631 interrupts or delivery errors. The default shall be **dead.letter** in the directory  
 23632 named by the *HOME* variable. The behavior of *mailx* in saving partial messages is  
 23633 unspecified if the User Portability Utilities option is not supported and *DEAD* is  
 23634 not defined with the value */dev/null*.

23635 **EDITOR** Determine the name of a utility to invoke when the **edit** (see **Commands in mailx**  
 23636 on page 629) or **~e** (see **Command Escapes in mailx** on page 637) command is  
 23637 XSI used. The default editor is *ed*. The effects of this variable are unspecified if the User  
 23638 Portability Utilities option is not supported.

23639 **HOME** Determine the path name of the user's home directory.

23640 **LANG** Provide a default value for the internationalization variables that are unset or null.  
 23641 If *LANG* is unset or null, the corresponding value from the implementation-  
 23642 dependent default locale shall be used. If any of the internationalization variables  
 23643 contains an invalid setting, the utility shall behave as if none of the variables had  
 23644 been defined.

23645 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 23646 internationalization variables.

23647 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 23648 characters (for example, single-byte as opposed to multi-byte characters in  
 23649 arguments and input files) and the handling of case-insensitive address and  
 23650 header-field comparisons.

23651 **LC\_TIME** Determine the format and contents of the date and time strings written by *mailx*.

## 23652 LC\_MESSAGES

23653 Determine the locale that should be used to affect the format and contents of  
 23654 diagnostic messages written to standard error and informative messages written to  
 23655 standard output.

23656 **LISTER** Determine a string representing the command for writing the contents of the  
 23657 **folder** directory to standard output when the **folders** command is given (see  
 23658 **folders** in **Commands in mailx** on page 629). Any string acceptable as a  
 23659 *command\_string* operand to the *sh -c* command shall be valid. If this variable is null  
 23660 or not set, the output command shall be *ls*. The effects of this variable are  
 23661 unspecified if the User Portability Utilities option is not supported.

23662 **MAILRC** Determine the path name of the start-up file. The default shall be **.mailrc** in the  
 23663 directory referred to by the *HOME* environment variable. The behavior of *mailx* is  
 23664 unspecified if the User Portability Utilities option is not supported and *MAILRC* is  
 23665 not defined with the value */dev/null*.

23666 **MBOX** Determine a path name of the file to save messages from the system mailbox that  
 23667 have been read. The **exit** command shall override this function, as shall saving the  
 23668 message explicitly in another file. The default shall be **mbox** in the directory

|       |                            |                                                                                                          |
|-------|----------------------------|----------------------------------------------------------------------------------------------------------|
| 23669 |                            | named by the <i>HOME</i> variable. The effects of this variable are unspecified if the                   |
| 23670 |                            | User Portability Utilities option is not supported.                                                      |
| 23671 | XSI                        | <b>NLSPATH</b> Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .     |
| 23672 |                            | <b>PAGER</b> Determine a string representing an output filtering or pagination command for               |
| 23673 |                            | writing the output to the terminal. Any string acceptable as a <i>command_string</i>                     |
| 23674 |                            | operand to the <i>sh -c</i> command shall be valid. When standard output is a terminal                   |
| 23675 |                            | device, the message output shall be piped through the command if the <i>mailx</i>                        |
| 23676 |                            | internal variable <b>crf</b> is set to a value less the number of lines in the message; see              |
| 23677 |                            | <b>Internal Variables in mailx</b> on page 626. If the <i>PAGER</i> variable is null or not set,         |
| 23678 |                            | the paginator shall be either <i>more</i> or another paginator utility documented in the                 |
| 23679 |                            | system documentation. The effects of this variable are unspecified if the User                           |
| 23680 |                            | Portability Utilities option is not supported.                                                           |
| 23681 |                            | <b>SHELL</b> Determine the name of a preferred command interpreter. The default shall be <i>sh</i> .     |
| 23682 |                            | The effects of this variable are unspecified if the User Portability Utilities option is                 |
| 23683 |                            | not supported.                                                                                           |
| 23684 |                            | <b>TERM</b> Determine the name of the terminal type, to indicate in an unspecified manner, if            |
| 23685 |                            | the internal variable <b>screen</b> is not specified, the number of lines in a screenful of              |
| 23686 |                            | headers. If <i>TERM</i> is not set or is set to null, an unspecified default terminal type               |
| 23687 |                            | shall be used and the value of a screenful is unspecified. The effects of this variable                  |
| 23688 |                            | are unspecified if the User Portability Utilities option is not supported.                               |
| 23689 |                            | <b>VISUAL</b> Determine a path name of a utility to invoke when the <b>visual</b> command (see           |
| 23690 |                            | <b>Commands in mailx</b> on page 629) or <i>~v</i> command-escape (see <b>Command Escapes</b>            |
| 23691 |                            | <b>in mailx</b> on page 637) is used. If this variable is null or not set, the full-screen               |
| 23692 |                            | editor shall be <i>vi</i> . The effects of this variable are unspecified if the User Portability         |
| 23693 |                            | Utilities option is not supported.                                                                       |
| 23694 | <b>ASYNCHRONOUS EVENTS</b> |                                                                                                          |
| 23695 |                            | When <i>mailx</i> is in Send Mode and standard input is not a terminal, it shall take the standard       |
| 23696 |                            | action for all signals.                                                                                  |
| 23697 |                            | In Receive Mode, or in Send Mode when standard input is a terminal, if a SIGINT signal is                |
| 23698 |                            | received:                                                                                                |
| 23699 |                            | 1. If in command mode, the current command, if there is one, shall be aborted, and a                     |
| 23700 |                            | command-mode prompt shall be written.                                                                    |
| 23701 |                            | 2. If in input mode:                                                                                     |
| 23702 |                            | a. If <b>ignore</b> is set, <i>mailx</i> shall write "@\n", discard the current input line, and continue |
| 23703 |                            | processing, bypassing the message-abort mechanism described in item 2b.                                  |
| 23704 |                            | b. If the interrupt was received while sending mail, either when in Receive Mode or in                   |
| 23705 |                            | Send Mode, a message shall be written, and another subsequent interrupt, with no                         |
| 23706 |                            | other intervening characters typed, shall be required to abort the mail message. If in                   |
| 23707 |                            | Receive Mode and another interrupt is received, a command-mode prompt shall be                           |
| 23708 |                            | written. If in Send Mode and another interrupt is received, <i>mailx</i> shall terminate with            |
| 23709 |                            | a non-zero status.                                                                                       |
| 23710 |                            | In both cases listed in item b, if the message is not empty:                                             |
| 23711 |                            | i. If <b>save</b> is enabled and the file named by <i>DEAD</i> can be created, the message               |
| 23712 |                            | shall be written to the file named by <i>DEAD</i> . If the file exists, the message shall                |
| 23713 |                            | be written to replace the contents of the file.                                                          |

23714                   ii. If **save** is not enabled, or the file named by *DEAD* cannot be created, the  
23715                   message shall not be saved.

23716           The *mailx* utility shall take the standard action for all other signals.

## 23717 **STDOUT**

23718           In command and input modes, all output, including prompts and messages, shall be written to  
23719           standard output.

## 23720 **STDERR**

23721           Used only for diagnostic messages.

## 23722 **OUTPUT FILES**

23723           Various *mailx* commands and command escapes can create or add to files, including the **mbox**,  
23724           the dead-letter file, and secondary mailboxes. When *mailx* is used as described in this volume of  
23725           IEEE Std. 1003.1-200x, these files shall be text files, formatted as follows:

23726           line beginning with **From**<space>  
23727           [one or more *header-lines*; see **Commands in mailx** on page 629]  
23728           *empty line*  
23729           [zero or more *body lines*  
23730           *empty line*]  
23731           [line beginning with **From**<space>...]

23732           where each message begins with the **From** <space> line shown, preceded by the beginning of  
23733           the file or an empty line. (The **From** <space> line is considered to be part of the message header,  
23734           but not one of the header-lines referred to in **Commands in mailx** on page 629; thus, it shall not  
23735           be affected by the **discard**, **ignore**, or **retain** commands.) The formats of the remainder of the  
23736           **From** <space> line and any additional header lines are unspecified, except that none shall be  
23737           empty. The format of a message body line is also unspecified, except that no line following an  
23738           empty line shall start with **From** <space>; *mailx* shall modify any such user-entered message  
23739           body lines (following an empty line and beginning with **From** <space>) by adding one or more  
23740           characters to precede the 'F'; it may add these characters to **From** <space> lines that are not  
23741           preceded by an empty line.

23742           When a message from the system mailbox or entered by the user is not a text file, it is  
23743           implementation-dependent how such a message is stored in files written by *mailx*.

## 23744 **EXTENDED DESCRIPTION**

23745           The entire Extended Description section shall apply only to implementations supporting the  
23746           User Portability Utilities option.

23747           The *mailx* utility cannot guarantee support for all character encodings in all circumstances. For  
23748           example, inter-system mail may be restricted to 7-bit data by the underlying network, 8-bit data  
23749           need not be portable to non-internationalized systems, and so on. Under these circumstances, it  
23750           is recommended that only characters defined in the ISO/IEC 646:1991 standard International  
23751           Reference Version (equivalent to ASCII) 7-bit range of characters be used.

23752           When *mailx* is invoked using one of the Receive Mode synopsis forms, it shall write a page of  
23753           header-summary lines (if **-N** was not specified and there are messages, see below), followed by  
23754           a prompt indicating that *mailx* can accept regular commands (see **Commands in mailx** on page  
23755           629); this is termed *command mode*. The page of header-summary lines shall contain the first new  
23756           message if there are new messages, or the first unread message if there are unread messages, or  
23757           the first message. When *mailx* is invoked using the Send Mode synopsis and standard input is a  
23758           terminal, if no subject is specified on the command line and the **asksub** variable is set, a prompt  
23759           for the subject shall be written. At this point, *mailx* is in input mode. This input mode is also  
23760           entered when using one of the Receive Mode synopsis forms and a reply or new message is

23761 composed using the **reply**, **Reply**, **followup**, **Followup**, or **mail** commands and standard input  
 23762 is a terminal. When the message is typed and the end of message is encountered, the message  
 23763 shall be passed to the mail delivery software. Commands can be entered by beginning a line  
 23764 with the escape character (by default, tilde ('~')) followed by a single command letter and  
 23765 optional arguments. See **Commands in mailx** on page 629 for a summary of these commands. It  
 23766 is unspecified what effect these commands will have if standard input is not a terminal when a  
 23767 message is entered using either the Send Mode synopsis, or the Read Mode commands **reply**,  
 23768 **Reply**, **followup**, **Followup**, or **mail**.

23769 **Note:** For notational convenience, this section uses the default escape character, tilde, in all  
 23770 references and examples.

23771 At any time, the behavior of *mailx* shall be governed by a set of environmental and internal  
 23772 variables. These are flags and valued parameters that can be set and cleared via the *mailx* **set**  
 23773 and **unset** commands.

23774 Regular commands are of the form:

23775 `[command] [msglist] [argument ...]`

23776 If no *command* is specified in command mode, **next** shall be assumed. In input mode, commands  
 23777 shall be recognized by the escape character, and lines not treated as commands shall be taken as  
 23778 input for the message.

23779 In command mode, each message shall be assigned a sequential number, starting with 1.

23780 All messages have a state that affects how they are displayed in the header summary and how  
 23781 they are retained or deleted upon termination of *mailx*. There is at any time the notion of a  
 23782 *current* message, marked by a '**>**' at the beginning of a line in the header summary. When *mailx*  
 23783 is invoked using one of the Receive Mode synopsis forms, the current message shall be the first  
 23784 new message, if there is a new message, or the first unread message if there is an unread  
 23785 message, or the first message if there are any messages, or unspecified if there are no messages  
 23786 in the mailbox. Each command that takes an optional list of messages (*msglist*) or an optional  
 23787 single message (*message*) on which to operate shall leave the current message set to the highest-  
 23788 numbered message of the messages specified, unless the command deletes messages, in which  
 23789 case the current message shall be set to the first undeleted message (that is, a message not in the  
 23790 deleted state) after the highest-numbered message deleted by the command, if one exists, or the  
 23791 first undeleted message before the highest-numbered message deleted by the command, if one  
 23792 exists, or to an unspecified value if there are no remaining undeleted messages. All messages are  
 23793 in one of the following states:

|       |               |                                                                                                                                                                                      |
|-------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 23794 | <i>new</i>    | The message is present in the system mailbox and has not been viewed by the user                                                                                                     |
| 23795 |               | or moved to any other state. Messages in state <i>new</i> when <i>mailx</i> quits shall be                                                                                           |
| 23796 |               | retained in the system mailbox.                                                                                                                                                      |
| 23797 | <i>unread</i> | The message has been present in the system mailbox for more than one invocation                                                                                                      |
| 23798 |               | of <i>mailx</i> and has not been viewed by the user or moved to any other state.                                                                                                     |
| 23799 |               | Messages in state <i>unread</i> when <i>mailx</i> quits shall be retained in the system mailbox.                                                                                     |
| 23800 | <i>read</i>   | The message has been processed by one of the following commands: <b>~f</b> , <b>~m</b> , <b>~F</b> , <b>~M</b> ,                                                                     |
| 23801 |               | <b>copy</b> , <b>mbox</b> , <b>next</b> , <b>pipe</b> , <b>print</b> , <b>Print</b> , <b>top</b> , <b>type</b> , <b>Type</b> , <b>undelete</b> . The <b>delete</b> , <b>dp</b> , and |
| 23802 |               | <b>dt</b> commands may also cause the next message to be marked as <i>read</i> , depending on                                                                                        |
| 23803 |               | the value of the <b>autoprint</b> variable. Messages that are in the system mailbox and in                                                                                           |
| 23804 |               | state <i>read</i> when <i>mailx</i> quits shall be saved in the <b>mbox</b> , unless the internal variable                                                                           |
| 23805 |               | <b>hold</b> was set. Messages that are in the <b>mbox</b> or in a secondary mailbox and in state                                                                                     |
| 23806 |               | <i>read</i> when <i>mailx</i> quits shall be retained in their current location.                                                                                                     |

|       |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 23807 | <i>deleted</i>   | The message has been processed by one of the following commands: <b>delete</b> , <b>dp</b> , <b>dt</b> . Messages in state <i>deleted</i> when <i>mailx</i> quits shall be deleted. Deleted messages shall be ignored until <i>mailx</i> quits or changes mailboxes or they are specified to the undelete command; for example, the message specification <i>/string</i> shall only search the subject lines of messages that have not yet been deleted, unless the command operating on the list of messages is <b>undelete</b> . No deleted message or deleted message header shall be displayed by any <i>mailx</i> command other than <b>undelete</b> . |
| 23808 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23809 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23810 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23811 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23812 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23813 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23814 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23815 | <i>preserved</i> | The message has been processed by a <b>preserve</b> command. When <i>mailx</i> quits, the message shall be retained in its current location.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 23816 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23817 | <i>saved</i>     | The message has been processed by one of the following commands: <b>save</b> or <b>write</b> . If the current mailbox is the system mailbox, and the internal variable <b>keepsave</b> is set, messages in the state <i>saved</i> shall be saved to the file designated by the <i>MBOX</i> variable (see the ENVIRONMENT VARIABLES section). If the current mailbox is the system mailbox, messages in the state <i>saved</i> shall be deleted from the current mailbox, when the <b>quit</b> or <b>file</b> command is used to exit the current mailbox.                                                                                                   |
| 23818 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23819 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23820 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23821 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23822 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23823 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23824 |                  | The header-summary line for each message shall indicate the state of the message.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 23825 |                  | Many commands take an optional list of messages ( <i>msglist</i> ) on which to operate, which defaults to the current message. A <i>msglist</i> is a list of message specifications separated by <blank> characters, which can include:                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 23826 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23827 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23828 | <i>n</i>         | Message number <i>n</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 23829 | <i>+</i>         | The next undeleted message, or the next deleted message for the <b>undelete</b> command.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 23830 | <i>-</i>         | The next previous undeleted message, or the next previous deleted message for the <b>undelete</b> command.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 23831 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23832 | <i>.</i>         | The current message.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 23833 | <i>^</i>         | The first undeleted message, or the first deleted message for the <b>undelete</b> command.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 23834 | <i>\$</i>        | The last message.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 23835 | <i>*</i>         | All messages.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 23836 | <i>n-m</i>       | An inclusive range of message numbers.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 23837 | <i>address</i>   | All messages from <i>address</i> ; any address as shown in a header summary shall be matchable in this form.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 23838 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23839 | <i>/string</i>   | All messages with <i>string</i> in the subject line (case ignored).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 23840 | <i>:c</i>        | All messages of type <i>c</i> , where <i>c</i> shall be one of:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23841 | <i>d</i>         | Deleted messages.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 23842 | <i>n</i>         | New messages.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 23843 | <i>o</i>         | Old messages (any not in state <i>read</i> or <i>new</i> ).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 23844 | <i>r</i>         | Read messages.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 23845 | <i>u</i>         | Unread messages.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 23846 |                  | Other commands take an optional message ( <i>message</i> ) on which to operate, which defaults to the current message. All of the forms allowed for <i>msglist</i> are also allowed for <i>message</i> , but if more than one message is specified, only the first shall be operated on.                                                                                                                                                                                                                                                                                                                                                                    |
| 23847 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 23848 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

23849 Other arguments are usually arbitrary strings whose usage depends on the command involved.

## 23850 Start-Up in mailx

23851 At start-up time, *mailx* shall take the following steps in sequence:

- 23852 1. Establish all variables at their stated default values.
- 23853 2. Process command line options, overriding corresponding default values.
- 23854 3. Import any of the *DEAD*, *EDITOR*, *MBOX*, *LISTER*, *PAGER*, *SHELL*, or *VISUAL* variables  
23855 that are present in the environment, overriding the corresponding default values.
- 23856 4. Read *mailx* commands from an unspecified system start-up file, unless the **-n** option is  
23857 given, to initialize any internal *mailx* variables and aliases.
- 23858 5. Process the start-up file of *mailx* commands named in the user *MAILRC* variable.

23859 Most regular *mailx* commands are valid inside start-up files, the most common use being to set  
23860 up initial display options and alias lists. The following commands shall be invalid in the start-up  
23861 file: **!**, **edit**, **hold**, **mail**, **preserve**, **reply**, **Reply**, **shell**, **visual**, **Copy**, **followup**, and **Followup**. Any  
23862 errors in the start-up file shall either cause *mailx* to terminate with a diagnostic message and a  
23863 non-zero status or to continue after writing a diagnostic message, ignoring the remainder of the  
23864 lines in the start-up file.

23865 A blank line in a start-up file shall be ignored.

## 23866 Internal Variables in mailx

23867 The following variables are internal *mailx* variables. Each internal variable can be set via the  
23868 *mailx set* command at any time. The **unset** and **set no name** commands can be used to erase  
23869 variables.

23870 In the following list, variables shown as:

23871 variable

23872 represent Boolean values. Variables shown as:

23873 variable=value

23874 shall be assigned string or numeric values. For string values, the rules in **Commands in mailx** on  
23875 page 629 concerning file names and quoting also apply.

23876 The defaults specified here may be changed by the implementation-dependent system start-up  
23877 file unless the user specifies the **-n** option.

23878 **allnet** All network names whose login name components match are treated as identical.  
23879 This shall cause the *msglist* message specifications to behave similarly. The default  
23880 shall be **noallnet**. See also the **alternates** command and the **metoo** variable.

23881 **append** Append messages to the end of the **mbox** file upon termination instead of placing  
23882 them at the beginning. The default shall be **noappend**. This variable shall not  
23883 affect the **save** command when saving to the **mbox**.

## 23884 ask, asksub

23885 Prompt for a subject line on outgoing mail if one is not specified on the command  
23886 line with the **-s** option. The **ask** and **asksub** forms are synonyms; the system shall  
23887 refer to **asksub** and **noasksub** in its messages, but shall accept **ask** and **noask** as  
23888 user input to mean **asksub** and **noasksub**. It shall not be possible to set both **ask**  
23889 and **noasksub**, or **noask** and **asksub**. The default shall be **asksub**, but no

|           |                         |                                                                                                                                                                                                                                                                                                                                                                        |
|-----------|-------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 23890     |                         | prompting shall be done if standard input is not a terminal.                                                                                                                                                                                                                                                                                                           |
| 23891     | <b>askbcc</b>           | Prompt for the blind copy list. The default shall be <b>noaskbcc</b> .                                                                                                                                                                                                                                                                                                 |
| 23892     | <b>askcc</b>            | Prompt for the copy list. The default shall be <b>noaskcc</b> .                                                                                                                                                                                                                                                                                                        |
| 23893     | <b>autoprint</b>        | Enable automatic writing of messages after <b>delete</b> and <b>undelete</b> commands. The default shall be <b>noautoprint</b> .                                                                                                                                                                                                                                       |
| 23894     |                         |                                                                                                                                                                                                                                                                                                                                                                        |
| 23895     | <b>bang</b>             | Enable the special-case treatment of exclamation-marks ( ' ! ' ) in escape command lines; see the <b>escape</b> command and <b>Command Escapes in mailx</b> on page 637. The default shall be <b>nobang</b> , disabling the expansion of ' ! ' in the <i>command</i> argument to the ~! command and the ~<!command escape.                                             |
| 23896     |                         |                                                                                                                                                                                                                                                                                                                                                                        |
| 23897     |                         |                                                                                                                                                                                                                                                                                                                                                                        |
| 23898     |                         |                                                                                                                                                                                                                                                                                                                                                                        |
| 23899     | <b>cmd=command</b>      |                                                                                                                                                                                                                                                                                                                                                                        |
| 23900     |                         | Set the default command to be invoked by the <b>pipe</b> command. The default shall be <b>nocmd</b> .                                                                                                                                                                                                                                                                  |
| 23901     |                         |                                                                                                                                                                                                                                                                                                                                                                        |
| 23902     | <b>crt=number</b>       | Pipe messages having more than <i>number</i> lines through the command specified by the value of the <i>PAGER</i> variable. The default shall be <b>nocrt</b> . If it is set to null, the value used is implementation-dependent.                                                                                                                                      |
| 23903     |                         |                                                                                                                                                                                                                                                                                                                                                                        |
| 23904     |                         |                                                                                                                                                                                                                                                                                                                                                                        |
| 23905 XSI | <b>debug</b>            | Enable verbose diagnostics for debugging. Messages are not delivered. The default shall be <b>nodebug</b> .                                                                                                                                                                                                                                                            |
| 23906     |                         |                                                                                                                                                                                                                                                                                                                                                                        |
| 23907     | <b>dot</b>              | When <b>dot</b> is set, a period on a line by itself during message input from a terminal shall also signify end-of-file (in addition to normal end-of-file). The default shall be <b>nodot</b> . If <i>ignoreeof</i> is set (see below), a setting of <b>nodot</b> shall be ignored and the period is the only method to terminate input mode.                        |
| 23908     |                         |                                                                                                                                                                                                                                                                                                                                                                        |
| 23909     |                         |                                                                                                                                                                                                                                                                                                                                                                        |
| 23910     |                         |                                                                                                                                                                                                                                                                                                                                                                        |
| 23911     | <b>escape=c</b>         | Set the command escape character to be the character ' c '. By default, the command escape character shall be tilde. If <b>escape</b> is unset, tilde shall be used; if it is set to null, command escaping shall be disabled.                                                                                                                                         |
| 23912     |                         |                                                                                                                                                                                                                                                                                                                                                                        |
| 23913     |                         |                                                                                                                                                                                                                                                                                                                                                                        |
| 23914     | <b>flipr</b>            | Reverse the meanings of the <b>R</b> and <b>r</b> commands. The default shall be <b>noflipr</b> .                                                                                                                                                                                                                                                                      |
| 23915     | <b>folder=directory</b> |                                                                                                                                                                                                                                                                                                                                                                        |
| 23916     |                         | The default directory for saving mail files. User-specified file names beginning with a plus sign ( ' + ' ) shall be expanded by preceding the file name with this directory name to obtain the real path name. If <i>directory</i> does not start with a slash ( ' / ' ), the contents of <i>HOME</i> shall be prefixed to it. The default shall be <b>nofolder</b> . |
| 23917     |                         | If <b>folder</b> is unset or set to null, user-specified file names beginning with ' + ' shall refer to files in the current directory that begin with the literal ' + ' character. See also <b>outfolder</b> below. The <b>folder</b> value need not affect the processing of the files named in <i>MBOX</i> and <i>DEAD</i> .                                        |
| 23918     |                         |                                                                                                                                                                                                                                                                                                                                                                        |
| 23919     |                         |                                                                                                                                                                                                                                                                                                                                                                        |
| 23920     |                         |                                                                                                                                                                                                                                                                                                                                                                        |
| 23921     |                         |                                                                                                                                                                                                                                                                                                                                                                        |
| 23922     |                         |                                                                                                                                                                                                                                                                                                                                                                        |
| 23923     |                         |                                                                                                                                                                                                                                                                                                                                                                        |
| 23924     | <b>header</b>           | Enable writing of the header summary when entering <i>mailx</i> in Receive Mode. The default shall be <b>header</b> .                                                                                                                                                                                                                                                  |
| 23925     |                         |                                                                                                                                                                                                                                                                                                                                                                        |
| 23926     | <b>hold</b>             | Preserve all messages that are read in the system mailbox instead of putting them in the <b>mbox</b> save file. The default shall be <b>nohold</b> .                                                                                                                                                                                                                   |
| 23927     |                         |                                                                                                                                                                                                                                                                                                                                                                        |
| 23928     | <b>ignore</b>           | Ignore interrupts while entering messages. The default shall be <b>noignore</b> .                                                                                                                                                                                                                                                                                      |
| 23929     | <i>ignoreeof</i>        |                                                                                                                                                                                                                                                                                                                                                                        |
| 23930     |                         | Ignore normal end-of-file during message input. Input can be terminated only by entering a period ( ' . ' ) on a line by itself or by the ~. command escape. The default shall be <b>noignoreeof</b> . See also <b>dot</b> above.                                                                                                                                      |
| 23931     |                         |                                                                                                                                                                                                                                                                                                                                                                        |
| 23932     |                         |                                                                                                                                                                                                                                                                                                                                                                        |

|           |                            |                                                                                                            |
|-----------|----------------------------|------------------------------------------------------------------------------------------------------------|
| 23933     | <b>indentprefix=string</b> |                                                                                                            |
| 23934     |                            | A string that shall be added as a prefix to each line that is inserted into the message                    |
| 23935     |                            | by the <b>~m</b> command escape. This variable shall default to one <tab> character.                       |
| 23936     | <b>keep</b>                | When a system mailbox, secondary mailbox, or <b>mbox</b> is empty, truncate it to zero                     |
| 23937     |                            | length instead of removing it. The default shall be <b>nokeep</b> .                                        |
| 23938     | <b>keepsave</b>            | Keep the messages that have been saved from the system mailbox into other files                            |
| 23939     |                            | in the file designated by the variable <i>MBOX</i> , instead of deleting them. The default                 |
| 23940     |                            | shall be <b>nokeepsave</b> .                                                                               |
| 23941     | <b>metoo</b>               | Suppress the deletion of the login name of the user from the recipient list when                           |
| 23942     |                            | replying to a message or sending to a group. The default shall be <b>nometoo</b> .                         |
| 23943 XSI | <b>onehop</b>              | When responding to a message that was originally sent to several recipients, the                           |
| 23944     |                            | other recipient addresses are normally forced to be relative to the originating                            |
| 23945     |                            | author's machine for the response. This flag disables alteration of the recipients'                        |
| 23946     |                            | addresses, improving efficiency in a network where all machines can send directly                          |
| 23947     |                            | to all other machines (that is, one hop away). The default shall be <b>noonehop</b> .                      |
| 23948     | <b>outfolder</b>           | Cause the files used to record outgoing messages to be located in the directory                            |
| 23949     |                            | specified by the <b>folder</b> variable unless the path name is absolute. The default shall                |
| 23950     |                            | be <b>nooutfolder</b> . See the <b>record</b> variable.                                                    |
| 23951     | <b>page</b>                | Insert a <form-feed> after each message sent through the pipe created by the <b>pipe</b>                   |
| 23952     |                            | command. The default shall be <b>nopage</b> .                                                              |
| 23953     | <b>prompt=string</b>       |                                                                                                            |
| 23954     |                            | Set the command-mode prompt to <i>string</i> . If <i>string</i> is null or if <b>noprompt</b> is set, no   |
| 23955     |                            | prompting shall occur. The default shall be to prompt with the string " ? ".                               |
| 23956     | <b>quiet</b>               | Refrain from writing the opening message and version when entering <i>mailx</i> . The                      |
| 23957     |                            | default shall be <b>noquiet</b> .                                                                          |
| 23958     | <b>record=file</b>         | Record all outgoing mail in the file with the path name <i>file</i> . The default shall be                 |
| 23959     |                            | <b>no record</b> . See also <b>outfolder</b> above.                                                        |
| 23960     | <b>save</b>                | Enable saving of messages in the dead-letter file on interrupt or delivery error. See                      |
| 23961     |                            | the variable <i>DEAD</i> for the location of the dead-letter file. The default shall be <b>save</b> .      |
| 23962     | <b>screen=number</b>       |                                                                                                            |
| 23963     |                            | Set the number of lines in a screenful of headers for the <b>headers</b> and <b>z</b> commands.            |
| 23964     |                            | If <b>screen</b> is not specified, a value based on the terminal type identified by the                    |
| 23965     |                            | <i>TERM</i> environment variable, the window size, the baud rate, or some combination                      |
| 23966     |                            | of these shall be used.                                                                                    |
| 23967 MAN | <b>sendwait</b>            | Wait for the background mailer to finish before returning. The default shall be                            |
| 23968     |                            | <b>no sendwait</b> .                                                                                       |
| 23969     | <b>showto</b>              | When the sender of the message was the user who is invoking <i>mailx</i> , write the                       |
| 23970     |                            | information from the <b>To:</b> line instead of the <b>From:</b> line in the header summary.               |
| 23971     |                            | The default shall be <b>noshowto</b> .                                                                     |
| 23972     | <b>sign=string</b>         | Set the variable inserted into the text of a message when the <b>~a</b> command escape is                  |
| 23973     |                            | given. The default shall be <b>no sign</b> . The character sequences ' <b>\t</b> ' and ' <b>\n</b> ' shall |
| 23974     |                            | be recognized in the variable as <tab> and <newline> characters, respectively. (See                        |
| 23975     |                            | also <b>~i</b> in <b>Command Escapes in mailx</b> on page 637.)                                            |
| 23976     | <b>Sign=string</b>         | Set the variable inserted into the text of a message when the <b>~A</b> command escape is                  |
| 23977     |                            | given. The default shall be <b>no Sign</b> . The character sequences ' <b>\t</b> ' and ' <b>\n</b> ' shall |



23978 be recognized in the variable as <tab> and <newline> characters, respectively.

23979 **toplines=number**

23980 Set the number of lines of the message to write with the **top** command. The default  
23981 shall be 5.

## 23982 **Commands in mailx**

23983 The following *mailx* commands shall be provided. In the following list, header refers to lines  
23984 from the message header, as shown in the OUTPUT FILES section. Header-line refers to lines  
23985 within the header that begin with one or more non-white-space characters, immediately  
23986 followed by a colon and white space and continuing until the next line beginning with a non-  
23987 white-space character or an empty line. Header-field refers to the portion of a header line prior  
23988 to the first colon in that line.

23989 For each of the commands listed below, the command can be entered as the abbreviation (those  
23990 characters in the Synopsis command word preceding the '[ ' '), the full command (all characters  
23991 shown for the command word, omitting the '[ ' ' and ' ] ' '), or any truncation of the full  
23992 command down to the abbreviation. For example, the **exit** command (shown as **ex[it]** in the  
23993 Synopsis) can be entered as **ex**, **exi**, or **exit**.

23994 The arguments to commands can be quoted, using the following methods:

- 23995 • An argument can be enclosed between paired double-quotes (" ") or single-quotes (" ' ' ");  
23996 any white space, shell word expansion, or backslash characters within the quotes shall be  
23997 treated literally as part of the argument. A double-quote shall be treated literally within  
23998 single-quotes and *vice versa*. These special properties of the quote marks shall occur only  
23999 when they are paired at the beginning and end of the argument.
- 24000 • A backslash outside of the enclosing quotes shall be discarded and the following character  
24001 treated literally as part of the argument.
- 24002 • An unquoted backslash at the end of a command line shall be discarded and the next line  
24003 shall continue the command.

24004 File names, where expected, shall be subjected to the process of shell word expansions (see  
24005 Section 2.6 on page 49); if more than a single path name results and the command is expecting  
24006 one file, the effects are unspecified. If the file name begins with an unquoted plus sign, it shall  
24007 not be expanded, but treated as the named file (less the leading plus) in the **folder** directory. (See  
24008 the **folder** variable.)

## 24009 **Declare Aliases**

24010 *Synopsis:*     a[lias] [alias [address...]]  
24011               g[roup] [alias [address...]]

24012 Add the given addresses to the alias specified by *alias*. The names shall be substituted when  
24013 *alias* is used as a recipient address specified by the user in an outgoing message (that is, other  
24014 recipients addressed indirectly through the **reply** command shall not be substituted in this  
24015 manner). Mail address alias substitution shall apply only when the alias string is used as a full  
24016 address; for example, when **hlj** is an alias, **hlj@posix.com** does not trigger the alias substitution. If  
24017 no arguments are given, write a listing of the current aliases to standard output. If only an *alias*  
24018 argument is given, write a listing of the specified alias to standard output. These listings need  
24019 not reflect the same order of addresses that were entered.

24020 **Declare Alternatives**

24021 *Synopsis:*     alt[ernates] *name...*

24022 (See also the **metoo** command.) Declare a list of alternative names for the user's login. When  
 24023 responding to a message, these names shall be removed from the list of recipients for the  
 24024 response. The comparison of names shall be in a case-insensitive manner. With no arguments,  
 24025 **alternates** shall write the current list of alternative names.

24026 **Change Current Directory**

24027 *Synopsis:*     cd [*directory*]  
 24028                 ch[dir] [*directory*]

24029 Change directory. If *directory* is not specified, the contents of *HOME* shall be used.

24030 **Copy Messages**

24031 *Synopsis:*     c[opy] [*file*]  
 24032                 c[opy] [*msglist*] *file*  
 24033 MAN           C[opy] [*msglist*]

24034 Copy messages to the file named by the path name *file* without marking the messages as saved.  
 24035 Otherwise, it shall be equivalent to the **save** command.

24036 MAN In the capitalized form, save the specified messages in a file whose name is derived from the  
 24037 author of the message to be saved, without marking the messages as saved. Otherwise, it shall  
 24038 be equivalent to the **Save** command.

24039 **Delete Messages**

24040 *Synopsis:*     d[ele]te [*msglist*]

24041 Mark messages for deletion from the mailbox. The deletions shall not occur until *mailx* quits (see  
 24042 the **quit** command) or changes mailboxes (see the **folder** command). If **autoprint** is set and there  
 24043 are messages remaining after the **delete** command, the current message shall be written as  
 24044 described for the **print** command (see the **print** command); otherwise, the *mailx* prompt shall be  
 24045 written.

24046 **Discard Header Fields**

24047 *Synopsis:*     di[scard] [*header-field...*]  
 24048                 ig[nore] [*header-field...*]

24049 Suppress the specified header fields when writing messages. Specified *header-fields* shall be  
 24050 added to the list of suppressed header fields. Examples of header fields to ignore are **status** and  
 24051 **cc**. The fields shall be included when the message is saved. The **Print** and **Type** commands shall  
 24052 override this command. The comparison of header fields shall be in a case-insensitive manner. If  
 24053 no arguments are specified, write a list of the currently suppressed header fields to standard  
 24054 output; the listing need not reflect the same order of header fields that were entered.

24055 If both **retain** and **discard** commands are given, **discard** commands shall be ignored.

## 24056 Delete Messages and Display

24057 *Synopsis:*    dp [*msglist*]  
 24058               dt [*msglist*]

24059 Delete the specified messages as described for the **delete** command, except that the **autoprint**  
 24060 variable shall have no effect, and the current message shall be written only if it was set to a  
 24061 message after the last message deleted by the command. Otherwise, an informational message  
 24062 to the effect that there are no further messages in the mailbox shall be written, followed by the  
 24063 *mailx* prompt.

## 24064 Echo a String

24065 *Synopsis:*    ec[ho] *string* ...

24066 Echo the given strings, equivalent to the shell *echo* utility.

## 24067 Edit Messages

24068 *Synopsis:*    e[dit] [*msglist*]

24069 Edit the given messages. The messages shall be placed in a temporary file and the utility named  
 24070 by the *EDITOR* variable is invoked to edit each file in sequence. The default *EDITOR* is  
 24071 unspecified.

24072 The **edit** command does not modify the contents of those messages in the mailbox.

## 24073 Exit

24074 *Synopsis:*    ex[it]  
 24075               x[it]

24076 Exit from *mailx* without changing the mailbox. No messages shall be saved in the **mbox** (see also  
 24077 **quit**).

## 24078 Change Folder

24079 *Synopsis:*    fi[le] [*file*]  
 24080               fold[er] [*file*]

24081 Quit (see the **quit** command) from the current file of messages and read in the file named by the  
 24082 path name *file*. If no argument is given, the name and status of the current mailbox shall be  
 24083 written.

24084 Several unquoted special characters shall be recognized when used as *file* names, with the  
 24085 following substitutions:

24086 %           The system mailbox for the invoking user.

24087 %*user*      The system mailbox for *user*.

24088 #           The previous file.

24089 &           The current **mbox**.

24090 +*file*      The named file in the **folder** directory. (See the **folder** variable.)

24091 The default file shall be the current mailbox.

24092 **Display List of Folders**24093 *Synopsis:*     **folders**

24094 Write the names of the files in the directory set by the **folder** variable. The command specified by  
 24095 the *LISTER* environment variable shall be used (see the ENVIRONMENT VARIABLES section).

24096 **Follow Up Specified Messages**24097 **Notes to Reviewers**24098 *This section with side shading will not appear in the final copy. - Ed.*

24099 D1, XCU, ERN 300 raises an issue re this text. An action item is outstanding.

24100 MAN *Synopsis:*     **fo**[llowup] [*message*]24101               **F**[ollowup] [*msglist*]

24102 In the lowercase form, respond to a message, recording the response in a file whose name is  
 24103 derived from the author of the message. Overrides the **record** variable, if set. See also the **save**  
 24104 and **copy** commands and **outfolder**.

24105 In the capitalized form, respond to the first message in the *msglist*, sending the message to the  
 24106 author of each message in the *msglist*. The subject line shall be taken from the first message and  
 24107 the response shall be recorded in a file whose name is derived from the author of the first  
 24108 message. See also the **Save** and **Copy** commands and **outfolder**.

24109 **Display Header Summary for Specified Messages**24110 *Synopsis:*     **f**[rom] [*msglist*]

24111 Write the header summary for the specified messages.

24112 **Display Header Summary**24113 *Synopsis:*     **h**[eaders] [*message*]

24114 Write the page of headers that includes the message specified. If the *message* argument is not  
 24115 specified, the current message shall not change. However, if the *message* argument is specified,  
 24116 the current message shall become the message that appears at the top of the page of headers that  
 24117 includes the message specified. The **screen** variable sets the number of headers per page. See  
 24118 also the **z** command.

24119 **Help**24120 *Synopsis:*     **hel**[p]

24121               ?

24122 Write a summary of commands.

24123 **Hold Messages**24124 *Synopsis:*     **ho**[ld] [*msglist*]24125               **pre**[serve] [*msglist*]

24126 Mark the messages in *msglist* to be retained in the mailbox when *mailx* terminates. This shall  
 24127 override any commands that might previously have marked the messages to be deleted. During  
 24128 the current invocation of *mailx*, only the **delete**, **dp**, or **dt** commands shall remove the *preserve*  
 24129 marking of a message.

24130        **Execute Commands Conditionally**

24131        *Synopsis:*     i[f] s|r  
 24132                    mail-commands  
 24133                    el[se]  
 24134                    mail-commands  
 24135                    en[dif]

24136        Execute commands conditionally, where **if s** executes the following *mail-commands*, up to an  
 24137        **else** or **endif**, if the program is in Send Mode, and **if r** shall cause the *mail-commands* to be  
 24138        executed only in Receive Mode.

24139        **List Available Commands**

24140        *Synopsis:*     l[ist]

24141        Write a list of all commands available. No explanation shall be given.

24142        **Mail a Message**

24143        *Synopsis:*     m[ail] address...

24144        Mail a message to the specified addresses or aliases.

24145        **Direct Messages to mbox**

24146        *Synopsis:*     mb[ox] [msglist]

24147        Arrange for the given messages to end up in the **mbox** save file when *mailx* terminates normally.  
 24148        See *MBOX*. See also the **exit** and **quit** commands.

24149        **Process Next Specified Message**

24150        *Synopsis:*     n[ext] [message]

24151        If the current message has not been written (for example, by the **print** command) since *mailx*  
 24152        started or since any other message was the current message, behave as if the **print** command  
 24153        was entered. Otherwise, if there is an undeleted message after the current message, make it the  
 24154        current message and behave as if the **print** command was entered. Otherwise, an informational  
 24155        message to the effect that there are no further messages in the mailbox shall be written, followed  
 24156        by the *mailx* prompt.

24157        **Pipe Message**

24158        *Synopsis:*     pi[pe] [[msglist] command]  
 24159                    | [[msglist] command]

24160        Pipe the messages through the given *command* by invoking the command interpreter specified  
 24161        by *SHELL* with two arguments: **-c** and *command*. (See also *sh -c*.) The application shall ensure  
 24162        that the command is given as a single argument. Quoting, described previously, can be used to  
 24163        accomplish this. If no arguments are given, the current message shall be piped through the  
 24164        command specified by the value of the **cmd** variable. If the **page** variable is set, a <form-feed>  
 24165        character shall be inserted after each message.

24166        **Display Message with Headers**

24167        *Synopsis:*     P[rint] [msglist]  
 24168                    T[ype] [msglist]

24169        Write the specified messages, including all header lines, to standard output. Override  
 24170        suppression of lines by the **discard**, **ignore**, and **retain** commands. If **crt** is set, the messages  
 24171        longer than the number of lines specified by the **crt** variable shall be paged through the  
 24172        command specified by the *PAGER* environment variable.

24173        **Display Message**

24174        *Synopsis:*     p[rint] [msglist]  
 24175                    t[ype] [msglist]

24176        Write the specified messages to standard output. If **crt** is set, the messages longer than the  
 24177        number of lines specified by the **crt** variable shall be paged through the command specified by  
 24178        the *PAGER* environment variable.

24179        **Quit**

24180        *Synopsis:*     q[uit]  
 24181                    end-of-file

24182        Terminate *mailx*, storing messages that were read in **mbox** (if the current mailbox is the system  
 24183        mailbox and unless **hold** is set), deleting messages that have been explicitly saved (unless  
 24184        **keepsave** is set), discarding messages that have been deleted, and saving all remaining messages  
 24185        in the mailbox.

24186        **Reply to a Message List**

24187        *Synopsis:*     R[eply] [msglist]  
 24188                    R[espond] [msglist]

24189        Mail a reply message to the sender of each message in the *msglist*. The subject line shall be  
 24190        formed by concatenating **Re:**<space> (unless it already begins with that string) and the subject  
 24191        from the first message. If **record** is set to a file name, the response shall be saved at the end of  
 24192        that file.

24193        See also the **flipr** variable.

24194        **Reply to a Message**

24195        *Synopsis:*     r[eply] [message]  
 24196                    r[espond] [message]

24197        Mail a reply message to all recipients included in the header of the message. The subject line  
 24198        shall be formed by concatenating **Re:**<space> (unless it already begins with that string) and the  
 24199        subject from the message. If **record** is set to a file name, the response shall be saved at the end of  
 24200        that file.

24201        See also the **flipr** variable.

24202      **Retain Header Fields**24203      *Synopsis:*      `ret[ain] [header-field...]`

24204      Retain the specified header fields when writing messages. This command shall override all  
 24205      **discard** and **ignore** commands. The comparison of header fields shall be in a case-insensitive  
 24206      manner. If no arguments are specified, write a list of the currently retained header fields to  
 24207      standard output; the listing need not reflect the same order of header fields that were entered.

24208      **Save Messages**

24209      *Synopsis:*      `s[ave] [file]`  
 24210                    `s[ave] [msglist] file`  
 24211 MAN            `S[ave] [msglist]`

24212      Save the specified messages in the file named by the path name *file*, or the **mbox** if the *file*  
 24213      argument is omitted. The file shall be created if it does not exist; otherwise, the messages shall be  
 24214      appended to the file. The message shall be put in the state *saved*, and shall behave as specified in  
 24215      the description of the *saved* state when the current mailbox is exited by the **quit** or **file**  
 24216      command.

24217 MAN      In the capitalized form, save the specified messages in a file whose name is derived from the  
 24218      author of the first message. The name of the file shall be taken to be the author's name with all  
 24219      network addressing stripped off. See also the **Copy**, **followup**, and **Followup** commands and  
 24220      **outfolder** variable.

24221      **Set Variables**24222      *Synopsis:*      `se[t] [name=[string]] ...] [name=number ...] [noname ...]`

24223      Define one or more variables called *name*. The variable can be given a null, string, or numeric  
 24224      value. Quoting and backslash escapes can occur anywhere in *string*, as described previously, as  
 24225      if the *string* portion of the argument were the entire argument. The forms *name* and *name*= shall  
 24226      be equivalent to *name*="" for variables that take string values. The **set** command without  
 24227      arguments shall write a list of all defined variables and their values. The **no name** form shall be  
 24228      equivalent to **unset name**.

24229      **Invoke a Shell**24230      *Synopsis:*      `sh[ell]`24231      Invoke an interactive command interpreter (see also *SHELL*).24232      **Display Message Size**24233      *Synopsis:*      `si[ze] [msglist]`

24234      Write the size in bytes of each of the specified messages.

24235      **Read mailx Commands From a File**24236      *Synopsis:*      `so[urce] file`

24237      Read and execute commands from the file named by the path name *file* and return to command  
 24238      mode.

### 24239 **Display Beginning of Messages**

24240 *Synopsis:*    to[p] [msglist]

24241 Write the top few lines of each of the specified messages. If the **toplines** variable is set, it is taken  
24242 as the number of lines to write. The default shall be 5.

### 24243 **Touch Messages**

24244 *Synopsis:*    tou[ch] [msglist]

24245 Touch the specified messages. If any message in *msglist* is not specifically deleted nor saved in a  
24246 file, it shall be placed in the **mbox** upon normal termination. See **exit** and **quit**.

### 24247 **Delete Aliases**

24248 *Synopsis:*    una[lias] [alias]...

24249 Delete the specified alias names. If a specified alias does not exist, the results are unspecified.

### 24250 **Undelete Messages**

24251 *Synopsis:*    u[ndelete] [msglist]

24252 Change the state of the specified messages from deleted to read. If **autoprint** is set, the last  
24253 message of those restored shall be written. If *msglist* is not specified, the message shall be  
24254 selected as follows:

- 24255       • If there are any deleted messages that follow the current message, the first of these shall be  
24256       chosen.
- 24257       • Otherwise, the last deleted message that also precedes the current message shall be chosen.

### 24258 **Unset Variables**

24259 *Synopsis:*    uns[et] name...

24260 Cause the specified variables to be erased.

### 24261 **Edit Message with Full-Screen Editor**

24262 *Synopsis:*    v[isual] [msglist]

24263 Edit the given messages with a screen editor. Each message shall be placed in a temporary file,  
24264 and the utility named by the *VISUAL* variable shall be invoked to edit each file in sequence. The  
24265 default editor shall be *vi*.

24266 The **visual** command does not modify the contents of those messages in the mailbox.

### 24267 **Write Messages to a File**

24268 *Synopsis:*    w[rite] [msglist] file

24269 Write the given messages to the file specified by the path name *file*, minus the message header.  
24270 Otherwise, it shall be equivalent to the **save** command.



24271 **Scroll Header Display**24272 *Synopsis:*      `z [+|-]`

24273 Scroll the header display forward (if '+' is specified or if no option is specified) or backward (if  
 24274 '-' is specified) one screenful. The number of headers written shall be set by the **screen**  
 24275 variable.

24276 **Invoke Shell Command**24277 *Synopsis:*      `!command`

24278 Invoke the command interpreter specified by *SHELL* with two arguments: `-c` and *command*.  
 24279 (See also *sh -c*.) If the **bang** variable is set, each unescaped occurrence of '!' in *command* shall  
 24280 be replaced with the command executed by the previous ! command or ~! command escape.

24281 **Null Command**24282 *Synopsis:*      `# comment`24283 This null command (comment) shall be ignored by *mailx*.24284 **Display Current Message Number**24285 *Synopsis:*      `=`

24286 Write the current message number.

24287 **Command Escapes in mailx**

24288 The following commands can be entered only from input mode, by beginning a line with the  
 24289 escape character (by default, tilde ('~')). See the **escape** variable description for changing this  
 24290 special character. The format for the commands shall be:

24291 `<ESC><command-char><separator>[<arguments>]`24292 where the `<separator>` can be zero or more `<blank>` characters.

24293 In the following descriptions, the application shall ensure that the argument *command* (but not  
 24294 *mailx-command*) is a shell command string. Any string acceptable to the command interpreter  
 24295 specified by the *SHELL* variable when it is invoked as *SHELL -c command\_string* shall be valid.  
 24296 The command can be presented as multiple arguments (that is, quoting is not required).

24297 Command escapes that are listed with *msglist* or *mailx-command* arguments are invalid in Send  
 24298 Mode and produce unspecified results.

24299 ~! *command* Invoke the command interpreter specified by *SHELL* with two arguments: `-c` and  
 24300 *command*; and then return to input mode. If the **bang** variable is set, each  
 24301 unescaped occurrence of '!' in *command* shall be replaced with the command  
 24302 executed by the previous ! command or ~! command escape.

24303 ~. Simulate end-of-file (terminate message input).

24304 ~: *mailx-command*, ~\_ *mailx-command*  
 24305 Perform the command-level request.

24306 ~? Write a summary of command escapes.

24307 ~A This shall be equivalent to ~i **Sign**.24308 ~a This shall be equivalent to ~i **sign**.

|       |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |
|-------|---------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 24309 | ~b <i>name</i> ...                                                              | Add the <i>names</i> to the blind carbon copy ( <b>Bcc</b> ) list.                                                                                                                                                                                                                                                                                                                       |
| 24310 | ~c <i>name</i> ...                                                              | Add the <i>names</i> to the carbon copy ( <b>Cc</b> ) list.                                                                                                                                                                                                                                                                                                                              |
| 24311 | ~d                                                                              | Read in the dead-letter file. See <i>DEAD</i> for a description of this file.                                                                                                                                                                                                                                                                                                            |
| 24312 | ~e                                                                              | Invoke the editor, as specified by the <i>EDITOR</i> environment variable, on the partial message.                                                                                                                                                                                                                                                                                       |
| 24313 |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |
| 24314 | ~f [ <i>msglist</i> ]                                                           | Forward the specified messages. The specified messages shall be inserted into the current message without alteration. This command escape also shall insert message headers into the message with field selection affected by the <b>discard</b> , <b>ignore</b> , and <b>retain</b> commands.                                                                                           |
| 24315 |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |
| 24316 |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |
| 24317 |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |
| 24318 | ~F [ <i>msglist</i> ]                                                           | This shall be the equivalent of the ~f command escape, except that all headers shall be included in the message, regardless of previous <b>discard</b> , <b>ignore</b> , and <b>retain</b> commands.                                                                                                                                                                                     |
| 24319 |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |
| 24320 |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |
| 24321 | ~h                                                                              | If standard input is a terminal, prompt for a <b>Subject</b> line and the <b>To</b> , <b>Cc</b> , and <b>Bcc</b> lists. Other implementation-dependent headers may also be presented for editing. If the field is written with an initial value, it can be edited as if it had just been typed.                                                                                          |
| 24322 |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |
| 24323 |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |
| 24324 |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |
| 24325 | ~i <i>string</i>                                                                | Insert the value of the named variable, followed by a <newline> character, into the text of the message. If the string is unset or null, the message shall not be changed.                                                                                                                                                                                                               |
| 24326 |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |
| 24327 | ~m [ <i>msglist</i> ]                                                           | Insert the specified messages into the message, prefixing non-empty lines with the string in the <b>indentprefix</b> variable. This command escape also shall insert message headers into the message, with field selection affected by the <b>discard</b> , <b>ignore</b> , and <b>retain</b> commands.                                                                                 |
| 24328 |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |
| 24329 |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |
| 24330 |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |
| 24331 | ~M [ <i>msglist</i> ]                                                           | This shall be the equivalent of the ~m command escape, except that all headers shall be included in the message, regardless of previous <b>discard</b> , <b>ignore</b> , and <b>retain</b> commands.                                                                                                                                                                                     |
| 24332 |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |
| 24333 |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |
| 24334 | ~p                                                                              | Write the message being entered. If the message is longer than <b>crt</b> lines (see <b>Internal Variables in mailx</b> on page 626), the output shall be paginated as described for the <i>PAGER</i> variable.                                                                                                                                                                          |
| 24335 |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |
| 24336 |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |
| 24337 | ~q                                                                              | Quit (see the <b>quit</b> command) from input mode by simulating an interrupt. If the body of the message is not empty, the partial message shall be saved in the dead-letter file. See <i>DEAD</i> for a description of this file.                                                                                                                                                      |
| 24338 |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |
| 24339 |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |
| 24340 | ""~r <i>file</i> , ~< <i>file</i> , ~r ! <i>command</i> , ~< ! <i>command</i> " |                                                                                                                                                                                                                                                                                                                                                                                          |
| 24341 |                                                                                 | Read in the file specified by the path name <i>file</i> . If the argument begins with an exclamation-mark (' ! '), the rest of the string shall be taken as an arbitrary system command; the command interpreter specified by <i>SHELL</i> shall be invoked with two arguments: <b>-c</b> and <i>command</i> . The standard output of <i>command</i> shall be inserted into the message. |
| 24342 |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |
| 24343 |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |
| 24344 |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |
| 24345 |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |
| 24346 | ~s <i>string</i>                                                                | Set the subject line to <i>string</i> .                                                                                                                                                                                                                                                                                                                                                  |
| 24347 | ~t <i>name</i> ...                                                              | Add the given <i>names</i> to the <b>To</b> list.                                                                                                                                                                                                                                                                                                                                        |
| 24348 | ~v                                                                              | Invoke the full-screen editor, as specified by the <i>VISUAL</i> environment variable, on the partial message.                                                                                                                                                                                                                                                                           |
| 24349 |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |
| 24350 | ~w <i>file</i>                                                                  | Write the partial message, without the header, onto the file named by the path name <i>file</i> . The file shall be created or the message shall be appended to it if the file exists.                                                                                                                                                                                                   |
| 24351 |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |
| 24352 |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                          |

24353       ~x           Exit as with ~q, except the message shall not be saved in the dead-letter file.

24354       ~| *command* Pipe the body of the message through the given *command* by invoking the  
 24355           command interpreter specified by *SHELL* with two arguments: -c and *command*.  
 24356           If the *command* returns a successful exit status, the standard output of the  
 24357           command shall replace the message. Otherwise, the message shall remain  
 24358           unchanged. If the *command* fails, an error message giving the exit status shall be  
 24359           written.

24360 **EXIT STATUS**

24361       When the -e option is specified, the following exit values are returned:

24362       0   Mail was found.

24363       >0   Mail was not found or an error occurred.

24364       Otherwise, the following exit values are returned:

24365       0   Successful completion; note that this status implies that all messages were *sent*, but it gives  
 24366           no assurances that any of them were actually *delivered*.

24367       >0   An error occurred.

24368 **CONSEQUENCES OF ERRORS**

24369       When in input mode (Receive Mode) or Send Mode:

24370       • If an error is encountered processing a command escape (see **Command Escapes in mailx** on  
 24371         page 637), a diagnostic message shall be written to standard error, and the message being  
 24372         composed may be modified, but this condition shall not prevent the message from being  
 24373         sent.

24374       • Other errors shall prevent the sending of the message.

24375       When in command mode:

24376       • Default.

24377 **APPLICATION USAGE**

24378       Delivery of messages to remote systems requires the existence of communication paths to such  
 24379       systems. These need not exist.

24380       Input lines are limited to {LINE\_MAX} bytes, but mailers between systems may impose more  
 24381       severe line-length restrictions. This volume of IEEE Std. 1003.1-200x does not place any  
 24382       restrictions on the length of messages handled by *mailx*, and for delivery of local messages the  
 24383       only limitations should be the normal problems of available disk space for the target mail file.  
 24384       When sending messages to external machines, applications are advised to limit messages to less  
 24385       than 100 kilobytes because some mail gateways impose message-length restrictions.

24386       The format of the system mailbox is intentionally unspecified. Not all systems implement  
 24387       system mailboxes as flat files, particularly with the advent of multimedia mail messages. Some  
 24388       system mailboxes may be multiple files, others records in a database. The internal format of the  
 24389       messages themselves are specified with the historical format from Version 7, but only after they  
 24390       have been saved in some file other than the system mailbox. This was done so that many  
 24391       historical applications expecting text-file mailboxes are not broken.

24392       Some new formats for messages can be expected in the future, probably including binary data,  
 24393       bit maps, and various multimedia objects. As described here, *mailx* is not prohibited from  
 24394       handling such messages, but it must store them as text files in secondary mailboxes (unless  
 24395       some extension, such as a variable or command line option, is used to change the stored format).  
 24396       Its method of doing so is implementation-dependent and might include translating the data into

24397 text file-compatible or readable form or omitting certain portions of the message from the stored  
24398 output.

24399 The **discard** and **ignore** commands are not inverses of the **retain** command. The **retain**  
24400 command discards all header-fields except those explicitly retained. The **discard** command  
24401 keeps all header-fields except those explicitly discarded. If headers exist on the retained header  
24402 list, **discard** and **ignore** commands are ignored.

#### 24403 EXAMPLES

24404 None.

#### 24405 RATIONALE

24406 The standard developers felt strongly that a method for applications to send messages to  
24407 specific users was necessary. The obvious example is a batch utility, running non-interactively,  
24408 that wishes to communicate errors or results to a user. However, the actual format, delivery  
24409 mechanism, and method of reading the message are clearly beyond the scope of this volume of  
24410 IEEE Std. 1003.1-200x.

24411 The intent of this command is to provide a simple, portable interface for sending messages non-  
24412 interactively. It merely defines a “front-end” to the historical mail system. It is suggested that  
24413 implementations explicitly denote the sender and recipient in the body of the delivered message.  
24414 Further specification of formats for either the message envelope or the message itself were  
24415 deliberately not made, as the industry is in the midst of changing from the current standards to a  
24416 more internationalized standard and it is probably incorrect, at this time, to require either one.

24417 Implementations are encouraged to conform to the various delivery mechanisms described in  
24418 the CCITT X.400 standards or to the equivalent Internet standards, described in Internet Request  
24419 for Comment (RFC) documents RFC 819, RFC 822, RFC 920, RFC 921, and RFC 1123.

24420 Many historical systems modified each body line that started with **From** by prefixing the ‘F’  
24421 with ‘>’. It is unnecessary, but allowed, to do that when the string does not follow a blank line  
24422 because it cannot be confused with the next header.

24423 XSI-conformant systems support the following internal variable:

24424 **debug** Enable verbose diagnostics for debugging. Messages shall not be delivered. The  
24425 default shall be **nodebug**.

24426 The *edit* and *visual* commands merely edit the specified messages in a temporary file. They do  
24427 not modify the contents of those messages in the mailbox; such a capability could be added as an  
24428 extension, such as by using different command names.

24429 The restriction on a subject line being {LINE\_MAX}–10 bytes is based on the historical format  
24430 that consumes 10 bytes for **Subject:** and the trailing <newline>. Many historical mailers that a  
24431 message may encounter on other systems are not able to handle lines that long, however.

24432 Like the utilities *logger* and *lp*, *mailx* admittedly is difficult to test. This was not deemed sufficient  
24433 justification to exclude this utility from this volume of IEEE Std. 1003.1-200x. It is also arguable  
24434 that it is, in fact, testable, but that the tests themselves are not portable.

24435 When *mailx* is being used by an application that wishes to receive the results as if none of the  
24436 User Portability Utilities option features were supported, the *DEAD* environment variable must  
24437 be set to **/dev/null**. Otherwise, it may be subject to the file creations described in *mailx*  
24438 ASYNCHRONOUS EVENTS. Similarly, if the *MAILRC* environment variable is not set to  
24439 **/dev/null**, historical versions of *mailx* and *Mail* read initialization commands from a file before  
24440 processing begins. Since the initialization that a user specifies could alter the contents of  
24441 messages an application is trying to send, such applications must set *MAILRC* to **/dev/null**.

24442 The description of *LC\_TIME* uses “may affect” because many historical implementations do not  
24443 or cannot manipulate the date and time strings in the incoming mail headers. Some headers  
24444 found in incoming mail do not have enough information to determine the timezone in which the  
24445 mail originated, and, therefore, *mailx* cannot convert the date and time strings into the internal  
24446 form that then is parsed by routines like *strftime()* that can take *LC\_TIME* settings into account.  
24447 Changing all these times to a user-specified format is allowed, but not required.

24448 The paginator selected when *PAGER* is null or unset is partially unspecified to allow the System  
24449 V historical practice of using *pg* as the default. Bypassing the pagination function, such as by  
24450 declaring that *cat* is the paginator, would not meet with the intended meaning of this  
24451 description. However, any “portable user” would have to set *PAGER* explicitly to get his or her  
24452 preferred paginator on all systems. The paginator choice was made partially unspecified, unlike  
24453 the *VISUAL* editor choice (mandated to be *vi*) because most historical pagers follow a common  
24454 theme of user input, whereas editors differ dramatically.

24455 Options to specify addresses as **cc** (carbon copy) or **bcc** (blind carbon copy) were considered to  
24456 be format details and were omitted.

24457 A zero exit status implies that all messages were *sent*, but it gives no assurances that any of them  
24458 were actually *delivered*. The reliability of the delivery mechanism is unspecified and is an  
24459 appropriate marketing distinction between systems.

24460 In order to conform to the Utility Syntax Guidelines, a solution was required to the optional *file*  
24461 option-argument to **-f**. By making *file* an operand, the guidelines are satisfied and users remain  
24462 portable. However, it does force implementations to support usage such as:

24463 `mailx -fin mymail.box`

24464 The **no name** method of unsetting variables is not present in all historical systems, but it is in  
24465 System V and provides a logical set of commands corresponding to the format of the display of  
24466 options from the *mailx set* command without arguments.

24467 The **ask** and **asksub** variables are the names selected by BSD and System V, respectively, for the  
24468 same feature. They are synonyms in this volume of IEEE Std. 1003.1-200x.

24469 The *mailx echo* command was not documented in the BSD version and has been omitted here  
24470 because it is not obviously useful for interactive users.

24471 The default prompt on the System V *mailx* is a question mark, on BSD *Mail* an ampersand. Since  
24472 this volume of IEEE Std. 1003.1-200x chose the *mailx* name, it kept the System V default,  
24473 assuming that BSD users would not have difficulty with this minor incompatibility (that they  
24474 can override).

24475 The meanings of **r** and **R** are reversed between System V *mailx* and SunOS *Mail*. Once again,  
24476 since this volume of IEEE Std. 1003.1-200x chose the *mailx* name, it kept the System V default,  
24477 but allows the SunOS user to achieve the desired results using **flipr**, an internal variable in  
24478 System V *mailx*, although it has not been documented in the SVID

24479 The **indentprefix** variable, the **retain** and **unalias** commands, and the **~F** and **~M** command  
24480 escapes were adopted from 4.3 BSD *Mail*.

24481 The **version** command was not included because no sufficiently general specification of the  
24482 version information could be devised that would still be useful to a portable user. This  
24483 command name should be used by suppliers who wish to provide version information about the  
24484 *mailx* command.

24485 The “implementation-specific (unspecified) system start-up file” historically has been named  
24486 **/etc/mailx.rc**, but this specific name and location are not required.

24487 The intent of the wording for the **next** command is that if any command has already displayed  
 24488 the current message it should display a following message, but, otherwise, it should display the  
 24489 current message. Consider the command sequence:

```
24490 next 3
24491 delete 3
24492 next
```

24493 where the **autoprint** option was not set. The normative text specifies that the second **next**  
 24494 command should display a message following the third message, because even though the  
 24495 current message has not been displayed since it was set by the **delete** command, it has been  
 24496 displayed since the current message was anything other than message number 3. This does not  
 24497 always match historical practice in some implementations, where the command file address  
 24498 followed by **next** (or the default command) would skip the message for which the user had  
 24499 searched.

#### 24500 FUTURE DIRECTIONS

24501 None.

#### 24502 SEE ALSO

24503 *ed, ls, more, vi*

#### 24504 CHANGE HISTORY

24505 First released in Issue 2.

#### 24506 Issue 4

24507 Aligned with the ISO/IEC 9945-2: 1993 standard.

24508 This utility is now mandatory; it is optional in Issue 3.

#### 24509 Issue 5

24510 The description of the EDITOR environment variable is changed to indicate that *ed* is the default  
 24511 editor if this variable is not set. In previous issues, this default was not stated explicitly at this  
 24512 point but was implied further down in the text.

24513 FUTURE DIRECTIONS section added.

#### 24514 Issue 6

24515 The following new requirements on POSIX implementations derive from alignment with the  
 24516 Single UNIX Specification:

- 24517 • The **-F** option is added.
- 24518 • The **allnet**, **debug**, and **sendwait** internal variables are added.
- 24519 • The **C**, **ec**, **fo**, **F**, and **S** *mailx* commands are added.

24520 In the DESCRIPTION and ENVIRONMENT VARIABLES sections, text stating “*HOME*  
 24521 directory” is replaced by “directory referred to by the *HOME* environment variable”.

24522 The *mailx* utility is aligned with the IEEE P1003.2b draft standard, which included various  
 24523 clarifications to resolve PASC Interpretations submitted for the ISO POSIX-2: 1993 standard. In  
 24524 particular, the changes here address PASC Interpretations 1003.2-92 #10, 11, 103, 106, 108, 114,  
 24525 115, 122, and 129.

24526 The normative text is reworded to avoid use of the term “must” for application requirements.

24527 **NAME**24528       **make** — maintain, update, and regenerate groups of programs (**DEVELOPMENT**)24529 **SYNOPSIS**

```
24530 SD make [-einpqrst][-f makefile...] [-k | -S][macro=value]...
24531 [target_name...]
```

24532

24533 **DESCRIPTION**

24534       The *make* utility can be used as a part of software development to update files that are derived  
 24535       from other files. A typical case is one where object files are derived from the corresponding  
 24536       source files. The *make* utility examines time relationships and updates those derived files (called  
 24537       targets) that have modified times earlier than the modified times of the files (called  
 24538       prerequisites) from which they are derived. A description file (makefile) contains a description  
 24539       of the relationships between files, and the commands that the application shall execute to  
 24540       update the targets to reflect changes in their prerequisites. Each specification, or rule, shall  
 24541       consist of a target, optional prerequisites, and optional commands to be executed when a  
 24542       prerequisite is newer than the target. There are two types of rule:

24543           1. *Inference rules*, which have one target name with at least one period ( ' . ' ) and no slash  
 24544            ( ' / ' )

24545           2. *Target rules*, which can have more than one target name

24546       In addition, *make* shall have a collection of built-in macros and inference rules that infer  
 24547       prerequisite relationships to simplify maintenance of programs.

24548       To receive exactly the behavior described in this section, the user shall ensure that a portable  
 24549       makefile:

- 24550       • Includes the special target **.POSIX**
- 24551       • Omits any special target reserved for implementations (a leading period followed by  
 24552        uppercase letters) that has not been specified by this section

24553       The behavior of *make* is unspecified if either or both of these conditions are not met.

24554 **OPTIONS**

24555       The *make* utility shall conform to the System Interface Definitions volume of  
 24556       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

24557       The following options shall be supported:

24558       **-e**           Cause environment variables, including those with null values, to override macro  
 24559       assignments within makefiles.

24560       **-f *makefile***   Specify a different makefile. The argument *makefile* is a path name of a description  
 24561       file, which is also referred to as the *makefile*. A path name of ' - ' shall denote the  
 24562       standard input. There can be multiple instances of this option, and they shall be  
 24563       processed in the order specified. The effect of specifying the same option-  
 24564       argument more than once is unspecified.

24565       **-i**           Ignore error codes returned by invoked commands. This mode is the same as if the  
 24566       special target **.IGNORE** were specified without prerequisites.

24567       **-k**           Continue to update other targets that do not depend on the current target if a non-  
 24568       ignored error occurs while executing the commands to bring a target up-to-date.

24569       **-n**           Write commands that would be executed on standard output, but do not execute  
 24570       them. However, lines with a plus sign ( ' + ' ) prefix shall be executed. In this mode,

24571 lines with an at sign ('@') character prefix shall be written to standard output.

24572       **-p**       Write to standard output the complete set of macro definitions and target  
24573       descriptions. The output format is unspecified.

24574       **-q**       Return a zero exit value if the target file is up-to-date; otherwise, return an exit  
24575       value of 1. Targets shall not be updated if this option is specified. However, a  
24576       makefile command line (associated with the targets) with a plus sign ('+') prefix  
24577       shall be executed.

24578       **-r**       Clear the suffix list and does not use the built-in rules.

24579       **-S**       Terminate *make* if an error occurs while executing the commands to bring a target  
24580       up-to-date. This shall be the default and the opposite of **-k**.

24581       **-s**       Do not write makefile command lines or touch messages (see **-t**) to standard  
24582       output before executing. This mode shall be the same as if the special target  
24583       **.SILENT** were specified without prerequisites.

24584       **-t**       Update the modification time of each target as though a *touch target* had been  
24585       executed. Targets that have prerequisites but no commands (see **Target Rules** on  
24586       page 647), or that are already up-to-date, shall not be touched in this manner.  
24587       Write messages to standard output for each target file indicating the name of the  
24588       file and that it was touched. Normally, the makefile command lines associated  
24589       with each target are not executed. However, a command line with a plus sign  
24590       ('+') prefix shall be executed.

24591       Any options specified in the *MAKEFLAGS* environment variable shall be evaluated before any  
24592       options specified on the *make* utility command line. If the **-k** and **-S** options are both specified  
24593       on the *make* utility command line or by the *MAKEFLAGS* environment variable, the last option  
24594       specified shall take precedence. If the **-f** or **-p** options appear in the *MAKEFLAGS* environment  
24595       variable, the result is undefined.

## 24596 OPERANDS

24597       The following operands shall be supported:

24598       *target\_name*   Target names, as defined in the EXTENDED DESCRIPTION section. If no target is  
24599       specified, while *make* is processing the makefiles, the first target that *make*  
24600       encounters that is not a special target or an inference rule shall be used.

24601       *macro=value*   Macro definitions, as defined in **Macros** on page 649.

24602       If the *target\_name* and *macro=value* operands are intermixed on the *make* utility command line,  
24603       the results are unspecified.

## 24604 STDIN

24605       The standard input shall be used only if the *makefile* option-argument is '-'. See the INPUT  
24606       FILES section.

## 24607 INPUT FILES

24608       The input file, otherwise known as the makefile, is a text file containing rules, macro definitions,  
24609       and comments.

## 24610 ENVIRONMENT VARIABLES

24611       The following environment variables shall affect the execution of *make*:

24612       **LANG**       Provide a default value for the internationalization variables that are unset or null.  
24613       If *LANG* is unset or null, the corresponding value from the implementation-  
24614       dependent default locale shall be used. If any of the internationalization variables  
24615       contains an invalid setting, the utility shall behave as if none of the variables had



|       |                    |                                                                                                      |
|-------|--------------------|------------------------------------------------------------------------------------------------------|
| 24616 |                    | been defined.                                                                                        |
| 24617 | <i>LC_ALL</i>      | If set to a non-empty string value, override the values of all the other                             |
| 24618 |                    | internationalization variables.                                                                      |
| 24619 | <i>LC_CTYPE</i>    | Determine the locale for the interpretation of sequences of bytes of text data as                    |
| 24620 |                    | characters (for example, single-byte as opposed to multi-byte characters in                          |
| 24621 |                    | arguments and input files).                                                                          |
| 24622 | <i>LC_MESSAGES</i> |                                                                                                      |
| 24623 |                    | Determine the locale that should be used to affect the format and contents of                        |
| 24624 |                    | diagnostic messages written to standard error.                                                       |
| 24625 | <i>MAKEFLAGS</i>   |                                                                                                      |
| 24626 |                    | This variable shall be interpreted as a character string representing a series of                    |
| 24627 |                    | option characters to be used as the default options. The implementation shall                        |
| 24628 |                    | accept both of the following formats (but need not accept them when intermixed):                     |
| 24629 |                    | • The characters are option letters without the leading hyphens or <blank>                           |
| 24630 |                    | character separation used on a <i>make</i> utility command line.                                     |
| 24631 |                    | • The characters are formatted in a manner similar to a portion of the <i>make</i> utility           |
| 24632 |                    | command line: options are preceded by hyphens and <blank> character-                                 |
| 24633 |                    | separated as described in the System Interface Definitions volume of                                 |
| 24634 |                    | IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines. The <i>macro=value</i>               |
| 24635 |                    | macro definition operands can also be included. The difference between the                           |
| 24636 |                    | contents of <i>MAKEFLAGS</i> and the <i>make</i> utility command line is that the contents           |
| 24637 |                    | of the variable shall not be subjected to the word expansions (see Section 2.6 on                    |
| 24638 |                    | page 49) associated with parsing the command line values.                                            |
| 24639 |                    | The value of the <i>SHELL</i> environment variable shall not be used as a macro and                  |
| 24640 |                    | shall not be modified by defining the <i>SHELL</i> macro in a makefile or on the <i>make</i>         |
| 24641 |                    | utility command line. All other environment variables, including those with null                     |
| 24642 |                    | values, shall be used as macros.                                                                     |
| 24643 | XSI                | <i>NLSPATH</i> Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> . |
| 24644 | XSI                | <i>PROJECTDIR</i>                                                                                    |
| 24645 |                    | Provide a directory to be used to search for SCCS files not found in the current                     |
| 24646 |                    | directory. In all of the following cases, the search for SCCS files is made in the                   |
| 24647 |                    | directory <b>SCCS</b> in the identified directory. If the value of <i>PROJECTDIR</i> begins          |
| 24648 |                    | with a slash, it shall be considered an absolute path name; otherwise, the value of                  |
| 24649 |                    | <i>PROJECTDIR</i> is treated as a user name and that user's initial working directory                |
| 24650 |                    | shall be examined for a subdirectory <b>src</b> or <b>source</b> . If such a directory is found, it  |
| 24651 |                    | shall be used. Otherwise, the value is used as a relative path name.                                 |
| 24652 |                    | If <i>PROJECTDIR</i> is not set or has a null value, the search for SCCS files shall be              |
| 24653 |                    | made in the directory <b>SCCS</b> in the current directory.                                          |
| 24654 |                    | The setting of <i>PROJECTDIR</i> affects all files listed in the remainder of this utility           |
| 24655 |                    | description for files with a component named <b>SCCS</b> .                                           |
| 24656 |                    | The value of the <i>SHELL</i> environment variable shall not be used as a macro and shall not be     |
| 24657 |                    | modified by defining the <b>SHELL</b> macro in a makefile or on the command line. All other          |
| 24658 |                    | environment variables, including those with null values, shall be used as macros, as defined in      |
| 24659 |                    | <b>Macros</b> on page 649.                                                                           |

24660 **ASYNCHRONOUS EVENTS**

24661 If not already ignored, *make* shall trap SIGHUP, SIGTERM, SIGINT, and SIGQUIT and remove  
 24662 the current target unless the target is a directory or the target is a prerequisite of the special  
 24663 target **.PRECIOUS** or unless one of the **-n**, **-p**, or **-q** options was specified. Any targets removed  
 24664 in this manner shall be reported in diagnostic messages of unspecified format, written to  
 24665 standard error. After this cleanup process, if any, *make* shall take the standard action for all other  
 24666 signals.

24667 **STDOUT**

24668 The *make* utility shall write all commands to be executed to standard output unless the **-s** option  
 24669 was specified, the command is prefixed with an at sign, or the special target **.SILENT** has either  
 24670 the current target as a prerequisite or has no prerequisites. If *make* is invoked without any work  
 24671 needing to be done, it shall write a message to standard output indicating that no action was  
 24672 taken. If the **-t** option is present and a file is touched, *make* shall write to standard output a  
 24673 message of unspecified format indicating that the file was touched, including the file name of  
 24674 the file.

24675 **STDERR**

24676 Used only for diagnostic messages.

24677 **OUTPUT FILES**

24678 Files can be created when the **-t** option is present. Additional files can also be created by the  
 24679 utilities invoked by *make*.

24680 **EXTENDED DESCRIPTION**

24681 The *make* utility attempts to perform the actions required to ensure that the specified targets are  
 24682 up-to-date. A target is considered out-of-date if it is older than any of its prerequisites or if it  
 24683 does not exist. The *make* utility shall treat all prerequisites as targets themselves and recursively  
 24684 ensure that they are up-to-date, processing them in the order in which they appear in the rule.  
 24685 The *make* utility shall use the modification times of files to determine whether the corresponding  
 24686 targets are out-of-date.

24687 After *make* has ensured that all of the prerequisites of a target are up-to-date and if the target is  
 24688 out-of-date, the commands associated with the target entry shall be executed. If there are no  
 24689 commands listed for the target, the target shall be treated as up-to-date.

24690 **Makefile Syntax**

24691 A makefile can contain rules, macro definitions (see **Macros** on page 649), and comments. There  
 24692 are two kinds of rules: *inference rules* and *target rules*. The *make* utility shall contain a set of  
 24693 built-in inference rules. If the **-r** option is present, the built-in rules shall not be used and the  
 24694 suffix list shall be cleared. Additional rules of both types can be specified in a makefile. If a rule  
 24695 is defined more than once, the value of the rule shall be that of the last one specified. Macros can  
 24696 also be defined more than once, and the value of the macro is specified in **Macros** on page 649.  
 24697 Comments start with a number sign (**#**) and continue until an unescaped <newline> character  
 24698 is reached.

24699 XSI By default, the following files shall be tried in sequence: **./makefile**, **./Makefile**, **./s.makefile**,  
 24700 **SCCS/s.makefile**, **./s.Makefile**, and **SCCS/s.Makefile**.

24701 The **-f** option shall direct *make* to ignore any of these default files and use the specified argument  
 24702 as a makefile instead. If the **-** argument is specified, standard input shall be used.

24703 The term *makefile* is used to refer to any rules provided by the user, whether in **./makefile** or its  
 24704 variants, or specified by the **-f** option.

24705 The rules in makefiles shall consist of the following types of lines: target rules, including special  
 24706 targets (see **Target Rules**), inference rules (see **Inference Rules** on page 650), macro definitions  
 24707 (see **Macros** on page 649), empty lines, and comments. Comments start with a number sign  
 24708 (`'#'`) and continue until an unescaped `<newline>` character is reached.

24709 When an escaped `<newline>` (one preceded by a backslash) is found anywhere in the makefile  
 24710 except in a command line, it shall be replaced, along with any leading white space on the  
 24711 following line, with a single `<space>`. When an escaped `<newline>` is found in a command line  
 24712 in a makefile, the command line shall contain the backslash, the `<newline>`, and the next line,  
 24713 except that the first character of the next line shall not be included if it is a `<tab>`.

## 24714 **Makefile Execution**

24715 Makefile command lines shall be processed one at a time by writing the makefile command line  
 24716 to the standard output (unless one of the conditions listed under `'@'` suppresses the writing)  
 24717 and executing the command(s) in the line. A `<tab>` character may precede the command to  
 24718 standard output. Command execution shall be as if the makefile command line were the  
 24719 argument to the `system()` function. The environment for the command being executed shall  
 24720 contain all of the variables in the environment of *make*.

24721 By default, when *make* receives a non-zero status from the execution of a command, it terminates  
 24722 with an error message to standard error.

24723 Makefile command lines can have one or more of the following prefixes: a hyphen (`'-'`), an at  
 24724 sign (`'@'`), or a plus sign (`'+'`). These modify the way in which *make* processes the command.  
 24725 When a command is written to standard output, the prefix shall not be included in the output.

24726 – If the command prefix contains a hyphen, or the `-i` option is present, or the special target  
 24727 `.IGNORE` has either the current target as a prerequisite or has no prerequisites, any error  
 24728 found while executing the command shall be ignored.

24729 @ If the command prefix contains an at sign and the *make* utility command line `-n` option is  
 24730 not specified, or the `-s` option is present, or the special target `.SILENT` has either the current  
 24731 target as a prerequisite or has no prerequisites, the command shall not be written to  
 24732 standard output before it is executed.

24733 + If the command prefix contains a plus sign, this indicates a makefile command line that  
 24734 shall be executed even if `-n`, `-q`, or `-t` is specified.

## 24735 **Target Rules**

24736 Target rules are formatted as follows:

```
24737 target [target...]: [prerequisite...][;command]
24738 [<tab>command
24739 <tab>command
24740 ...]
```

24741 *line that does not begin with `<tab>`*

24742 Target entries are specified by a `<blank>` character-separated, non-null list of targets, then a  
 24743 colon, then a `<blank>` character-separated, possibly empty list of prerequisites. Text following a  
 24744 semicolon, if any, and all following lines that begin with a `<tab>` character, are makefile  
 24745 command lines to be executed to update the target. The first non-empty line that does not begin  
 24746 with a `<tab>` character or `'#'` shall begin a new entry. An empty or blank line, or a line  
 24747 beginning with `'#'`, may begin a new entry.

24748 Applications shall select target names from the set of characters consisting solely of periods,  
 24749 underscores, digits, and alphabets from the portable character set (see the System Interface  
 24750 Definitions volume of IEEE Std. 1003.1-200x, Section 6.1, Portable Character Set).  
 24751 Implementations may allow other characters in target names as extensions. The interpretation of  
 24752 targets containing the characters '%' and ' ' is implementation-dependent.

24753 A target that has prerequisites, but does not have any commands, can be used to add to the  
 24754 prerequisite list for that target. Only one target rule for any given target can contain commands.

24755 Lines that begin with one of the following are called *special targets* and control the operation of  
 24756 *make*:

24757 **.DEFAULT** If the makefile uses this special target, the application shall ensure that it is  
 24758 specified with commands, but without prerequisites. The commands shall be used  
 24759 by *make* if there are no other rules available to build a target.

24760 **.IGNORE** Prerequisites of this special target are targets themselves; this shall cause errors  
 24761 from commands associated with them to be ignored in the same manner as  
 24762 specified by the **-i** option. Subsequent occurrences of **.IGNORE** shall add to the  
 24763 list of targets ignoring command errors. If no prerequisites are specified, *make* shall  
 24764 behave as if the **-i** option had been specified and errors from all commands  
 24765 associated with all targets shall be ignored.

24766 **.POSIX** The application shall ensure that this special target is specified without  
 24767 prerequisites or commands. If it appears before the first non-comment line in the  
 24768 makefile, *make* shall process the makefile as specified by this section; otherwise, the  
 24769 behavior of *make* is unspecified.

24770 **.PRECIOUS** Prerequisites of this special target shall not be removed if *make* receives one of the  
 24771 asynchronous events explicitly described in the ASYNCHRONOUS EVENTS  
 24772 section. Subsequent occurrences of **.PRECIOUS** shall add to the list of precious  
 24773 files. If no prerequisites are specified, all targets in the makefile shall be treated as  
 24774 if specified with **.PRECIOUS**.

24775 XSI **.SCCS\_GET** The application shall ensure that this special target is specified without  
 24776 prerequisites. If this special target is included in a makefile, the commands  
 24777 specified with this target shall replace the default commands associated with this  
 24778 special target (see **Default Rules** on page 653). The commands specified with this  
 24779 target are used to get all SCCS files that are not found in the current directory.

24780 When source files are named in a dependency list, *make* treats them just like any  
 24781 other target. Because the source file is presumed to be present in the directory,  
 24782 there is no need to add an entry for it to the makefile. When a target has no  
 24783 dependencies, but is present in the directory, *make* assumes that that file is up-to-  
 24784 date. If, however, an SCCS file named **SCCS/s.source\_file** is found for a target  
 24785 *source\_file*, *make* does some additional checking to assure that the target is up-to-  
 24786 date. If the target is missing, or if the SCCS file is newer, *make* automatically issues  
 24787 the commands specified for the **.SCCS\_GET** special target to retrieve the most  
 24788 recent version. However, if the target is writable by anyone, *make* does not retrieve  
 24789 a new version.

24790 **.SILENT** Prerequisites of this special target are targets themselves; this shall cause  
 24791 commands associated with them to not be written to the standard output before  
 24792 they are executed. Subsequent occurrences of **.SILENT** shall add to the list of  
 24793 targets with silent commands. If no prerequisites are specified, *make* shall behave  
 24794 as if the **-s** option had been specified and no commands or touch messages  
 24795 associated with any target shall be written to standard output.

24796 **.SUFFIXES** Prerequisites of **.SUFFIXES** shall be appended to the list of known suffixes and are  
 24797 used in conjunction with the inference rules (see **Inference Rules** on page 650). If  
 24798 **.SUFFIXES** does not have any prerequisites, the list of known suffixes shall be  
 24799 cleared.

24800 The special targets **.IGNORE**, **.POSIX**, **.PRECIOUS**, **.SILENT**, and **.SUFFIXES** shall be specified  
 24801 without commands.

24802 Targets with names consisting of a leading period followed by the uppercase letters "POSIX"  
 24803 and then any other characters are reserved for future standardization. Targets with names  
 24804 consisting of a leading period followed by one or more uppercase letters are reserved for  
 24805 implementation extensions.

## 24806 **Macros**

24807 Macro definitions are in the form:

24808 *string1* = [*string2*]

24809 The macro named *string1* is defined as having the value of *string2*, where *string2* is defined as all  
 24810 characters, if any, after the equal sign, up to a comment character ('#') or an unescaped  
 24811 <newline> character. Any <blank> characters immediately before or after the equal sign shall be  
 24812 ignored.

24813 Applications shall select macro names from the set of characters consisting solely of periods,  
 24814 underscores, digits, and alphabets from the portable character set (see the System Interface  
 24815 Definitions volume of IEEE Std. 1003.1-200x, Section 6.1, Portable Character Set). A macro name  
 24816 shall not contain an equals sign. Implementations may allow other characters in macro names  
 24817 as extensions.

24818 Macros can appear anywhere in the makefile.  $\$(string1)$  or  $\${string1}$  shall be replaced by  
 24819 *string2*, as follows:

- 24820 • Macros in target lines shall be evaluated when the target line is read.
- 24821 • Macros in makefile command lines shall be evaluated when the command is executed.
- 24822 • Macros in the string before the equals sign in a macro definition shall be evaluated when the  
 24823 macro assignment is made.
- 24824 • Macros after the equals sign in a macro definition shall not be evaluated until the defined  
 24825 macro is used in a rule or command, or before the equals sign in a macro definition.

24826 The parentheses or braces are optional if *string1* is a single character. The macro **\$\$** shall be  
 24827 replaced by the single character '\$'.

24828 The forms  $\$(string1[:subst1=[subst2]])$  or  $\${string1[:subst1=[subst2]]}$  can be used to replace all  
 24829 occurrences of *subst1* with *subst2* when the macro substitution is performed. The *subst1* to be  
 24830 replaced shall be recognized when it is a suffix at the end of a word in *string1* (where a *word*, in  
 24831 this context, is defined to be a string delimited by the beginning of the line, a <blank> or  
 24832 <newline> character).

24833 Macro definitions shall be taken from the following sources, in the following logical order,  
 24834 before the makefile(s) are read.

- 24835 1. Macros specified on the *make* utility command line, in the order specified on the command  
 24836 line. It is unspecified whether the internal macros defined in **Internal Macros** on page 652  
 24837 are accepted from this source.

2. Macros defined by the *MAKEFLAGS* environment variable, in the order specified in the environment variable. It is unspecified whether the internal macros defined in **Internal Macros** on page 652 are accepted from this source.
3. The contents of the environment, excluding the *MAKEFLAGS* and *SHELL* variables and including the variables with null values.
4. Macros defined in the inference rules built into *make*.

Macro definitions from these sources shall not override macro definitions from a lower-numbered source. Macro definitions from a single source (for example, the *make* utility command line, the *MAKEFLAGS* environment variable, or the other environment variables) shall override previous macro definitions from the same source.

Macros defined in the makefile(s) shall override macro definitions that occur before them in the makefile(s) and macro definitions from source 4. If the *-e* option is not specified, macros defined in the makefile(s) shall override macro definitions from source 3. Macros defined in the makefile(s) shall not override macro definitions from source 1. or source 2.

Before the makefile(s) are read, all of the *make* utility command line options (except *-f* and *-p*) and *make* utility command line macro definitions (except any for the *MAKEFLAGS* macro), not already included in the *MAKEFLAGS* macro, shall be added to the *MAKEFLAGS* macro. Other implementation-dependent options and macros may also be added to the *MAKEFLAGS* macro. If this modifies the value of the *MAKEFLAGS* macro, or, if the *MAKEFLAGS* macro is modified at any subsequent time, the *MAKEFLAGS* environment variable shall be modified to match the new value of the *MAKEFLAGS* macro.

Before the makefile(s) are read, all of the *make* utility command line macro definitions (except the *MAKEFLAGS* macro or the *SHELL* macro) shall be added to the environment of *make*. Other implementation-dependent variables may also be added to the environment of *make*.

The **SHELL** macro shall be treated specially. It shall be provided by *make* and set to the path name of the shell command language interpreter (see *sh* on page 888). The *SHELL* environment variable shall not affect the value of the **SHELL** macro. If **SHELL** is defined in the makefile or is specified on the command line, it shall replace the original value of the **SHELL** macro, but shall not affect the *SHELL* environment variable. Other effects of defining **SHELL** in the makefile or on the command line are implementation-dependent.

## Inference Rules

Inference rules are formatted as follows:

```
target:
<tab>command
[<tab>command]
...

line that does not begin with <tab> or #
```

The application shall ensure that the *target* portion is a valid target name (see **Target Rules** on page 647) of the form *.s2* or *.s1.s2* (where *.s1* and *.s2* are suffixes that have been given as prerequisites of the **.SUFFIXES** special target and *s1* and *s2* do not contain any slashes or periods.) If there is only one period in the target, it is a single-suffix inference rule. Targets with two periods are double-suffix inference rules. Inference rules can have only one target before the colon.

The application shall ensure that the makefile does not specify prerequisites for inference rules; no characters other than white space shall follow the colon in the first line, except when creating

24883 the *empty rule*, described below. Prerequisites are inferred, as described below.

24884 Inference rules can be redefined. A target that matches an existing inference rule shall overwrite  
 24885 the old inference rule. An empty rule can be created with a command consisting of simply a  
 24886 semicolon (that is, the rule still exists and is found during inference rule search, but since it is  
 24887 empty, execution has no effect). The empty rule also can be formatted as follows:

24888 `rule: ;`

24889 where zero or more <blank> characters separate the colon and semicolon.

24890 The *make* utility uses the suffixes of targets and their prerequisites to infer how a target can be  
 24891 made up-to-date. A list of inference rules defines the commands to be executed. By default, *make*  
 24892 contains a built-in set of inference rules. Additional rules can be specified in the makefile.

24893 The special target **.SUFFIXES** contains as its prerequisites a list of suffixes that shall be used by  
 24894 the inference rules. The order in which the suffixes are specified defines the order in which the  
 24895 inference rules for the suffixes are used. New suffixes shall be appended to the current list by  
 24896 specifying a **.SUFFIXES** special target in the makefile. A **.SUFFIXES** target with no prerequisites  
 24897 shall clear the list of suffixes. An empty **.SUFFIXES** target followed by a new **.SUFFIXES** list is  
 24898 required to change the order of the suffixes.

24899 Normally, the user would provide an inference rule for each suffix. The inference rule to update  
 24900 a target with a suffix **.s1** from a prerequisite with a suffix **.s2** is specified as a target **.s2.s1**. The  
 24901 internal macros provide the means to specify general inference rules (see **Internal Macros** on  
 24902 page 652).

24903 When no target rule is found to update a target, the inference rules shall be checked. The suffix  
 24904 of the target (**.s1**) to be built is compared to the list of suffixes specified by the **.SUFFIXES** special  
 24905 targets. If the **.s1** suffix is found in **.SUFFIXES**, the inference rules shall be searched in the order  
 24906 defined for the first **.s2.s1** rule whose prerequisite file (**\$\*.s2**) exists. If the target is out-of-date  
 24907 with respect to this prerequisite, the commands for that inference rule shall be executed.

24908 If the target to be built does not contain a suffix and there is no rule for the target, the single  
 24909 suffix inference rules shall be checked. The single-suffix inference rules define how to build a  
 24910 target if a file is found with a name that matches the target name with one of the single suffixes  
 24911 appended. A rule with one suffix **.s2** is the definition of how to build *target* from **target.s2**. The  
 24912 other suffix (**.s1**) is treated as null.

24913 XSI A tilde ('~') in the above rules refers to an SCCS file in the current directory. Thus, the rule **.c~.o**  
 24914 would transform an SCCS C-language source file into an object file (**.o**). Because the **s.** of the  
 24915 SCCS files is a prefix, it is incompatible with *make*'s suffix point of view. Hence, the '~' is a way  
 24916 of changing any file reference into an SCCS file reference.

24917 **Libraries**

24918 If a target or prerequisite contains parentheses, it shall be treated as a member of an archive  
 24919 library. For the *lib(member.o)* expression *lib* refers to the name of the archive library and *member.o*  
 24920 to the member name. The application shall ensure that the member is an object file with the **.o**  
 24921 suffix. The modification time of the expression is the modification time for the member as kept  
 24922 in the archive library; see *ar* on page 168. The **.a** suffix refers to an archive library. The **.s2.a** rule  
 24923 is used to update a member in the library from a file with a suffix **.s2**.

24924      **Internal Macros**

24925      The *make* utility shall maintain five internal macros that can be used in target and inference rules.  
 24926      In order to clearly define the meaning of these macros, some clarification of the terms *target rule*,  
 24927      *inference rule*, *target*, and *prerequisite* is necessary.

24928      Target rules are specified by the user in a makefile for a particular target. Inference rules are  
 24929      user-specified or *make*-specified rules for a particular class of target name. Explicit prerequisites  
 24930      are those prerequisites specified in a makefile on target lines. Implicit prerequisites are those  
 24931      prerequisites that are generated when inference rules are used. Inference rules are applied to  
 24932      implicit prerequisites or to explicit prerequisites that do not have target rules defined for them in  
 24933      the makefile. Target rules are applied to targets specified in the makefile.

24934      Before any target in the makefile is updated, each of its prerequisites (both explicit and implicit)  
 24935      shall be updated. This shall be accomplished by recursively processing each prerequisite. Upon  
 24936      recursion, each prerequisite shall become a target itself. Its prerequisites in turn shall be  
 24937      processed recursively until a target is found that has no prerequisites, at which point the  
 24938      recursion stops. The recursion then shall back up, updating each target as it goes.

24939      In the definitions that follow, the word *target* refers to one of:

- 24940      • A target specified in the makefile
- 24941      • An explicit prerequisite specified in the makefile that becomes the target when *make*  
 24942      processes it during recursion
- 24943      • An implicit prerequisite that becomes a target when *make* processes it during recursion

24944      In the definitions that follow, the word *prerequisite* refers to one of the following:

- 24945      • An explicit prerequisite specified in the makefile for a particular target
- 24946      • An implicit prerequisite generated as a result of locating an appropriate inference rule and  
 24947      corresponding file that matches the suffix of the target

24948      The five internal macros are:

24949      **\$@**      The **\$@** shall evaluate to the full target name of the current target, or the archive file  
 24950      name part of a library archive target. It shall be evaluated for both target and inference  
 24951      rules.

24952      For example, in the **.c.a** inference rule, **\$@** represents the out-of-date **.a** file to be built.  
 24953      Similarly, in a makefile target rule to build **lib.a** from **file.c**, **\$@** represents the out-of-  
 24954      date **lib.a**.

24955      **\$%**      The **\$%** macro shall be evaluated only when the current target is an archive library  
 24956      member of the form *libname(member.o)*. In these cases, **\$@** shall evaluate to *libname* and  
 24957      **\$%** shall evaluates to *member.o*. The **\$%** macro shall be evaluated for both target and  
 24958      inference rules.

24959      For example, in a makefile target rule to build **lib.a(file.o)**, **\$%** represents **file.o**, as  
 24960      opposed to **\$@**, which represents **lib.a**.

24961      **\$?**      The **\$?** macro shall evaluate to the list of prerequisites that are newer than the current  
 24962      target. It shall be evaluated for both target and inference rules.

24963      For example, in a makefile target rule to build *prog* from **file1.o**, **file2.o**, and **file3.o**, and  
 24964      where *prog* is not out of date with respect to **file1.o**, but is out of date with respect to  
 24965      **file2.o** and **file3.o**, **\$?** represents **file2.o** and **file3.o**.



24966       \$<       In an inference rule, the \$< macro shall evaluate to the file name whose existence  
 24967               allowed the inference rule to be chosen for the target. In the **.DEFAULT** rule, the \$<  
 24968               macro shall evaluate to the current target name. The meaning of the \$< macro is  
 24969               otherwise unspecified.

24970               For example, in the **.c.a** inference rule, \$< represents the prerequisite **.c** file.

24971       \$\*       The \$\* macro shall evaluate to the current target name with its suffix deleted. It shall be  
 24972               evaluated at least for inference rules.

24973               For example, in the **.c.a** inference rule, \$\*.o represents the out-of-date **.o** file that  
 24974               corresponds to the prerequisite **.c** file.

24975       Each of the internal macros has an alternative form. When an uppercase 'D' or 'F' is appended  
 24976       to any of the macros, the meaning is changed to the *directory part* for 'D' and *file name part* for  
 24977       'F'. The directory part is the path prefix of the file without a trailing slash; for the current  
 24978       directory, the directory part is ' '. When the \$? macro contains more than one prerequisite file  
 24979       name, the \$(?D) and \$(?F) (or \${?D} and \${?F}) macros expand to a list of directory name parts  
 24980       and file name parts respectively.

24981       For the target *lib(member.o)* and the **s2.a** rule, the internal macros are defined as:

24982       \$<       *member.s2*

24983       \$\*       *member*

24984       \$@       *lib*

24985       \$?       *member.s2*

24986       \$%       *member.o*

## 24987       Default Rules

24988       The default rules for *make* shall achieve results that are the same as if the following were used.  
 24989       Implementations that do not support the C-Language Development Utilities option may omit  
 24990       **CC**, **CFLAGS**, **YACC**, **YFLAGS**, **LEX**, **LFLAGS**, **LDFLAGS**, and the **.c**, **.y**, and **.l** inference rules.  
 24991       Implementations that do not support FORTRAN may omit **FC**, **FFLAGS**, and the **.f** inference  
 24992       rules. Implementations may provide additional macros and rules.

## 24993       SPECIAL TARGETS

24994 XSI       .SCCS\_GET: sccs \$(SCCSFLAGS) get \$(SCCSGETFLAGS) \$@

24995

24996 XSI       .SUFFIXES: .o .c .y .l .a .sh .f .c~ .y~ .l~ .sh~ .f~

## 24997       MACROS

24998       MAKE=make

24999       AR=ar

25000       ARFLAGS=-rv

25001       YACC=yacc

25002       YFLAGS=

25003       LEX=lex

25004       LFLAGS=

25005       LDFLAGS=

25006       CC=c89

25007       CFLAGS=-O

25008       FC=fort77

```

25009 FFLAGS=-O 1
25010 XSI GET=get
25011 GFLAGS=
25012 SCCSFLAGS=
25013 SCCSGETFLAGS=-s
25014

```

#### 25015 *SINGLE SUFFIX RULES*

```

25016 .c:
25017 $(CC) $(CFLAGS) $(LDFLAGS) -o $@ $<
25018 .f:
25019 $(FC) $(FFLAGS) $(LDFLAGS) -o $@ $<
25020 .sh:
25021 cp $< $@
25022 chmod a+x $@

```

```

25023 XSI .c~:
25024 $(GET) $(GFLAGS) -p $< > $*.c
25025 $(CC) $(CFLAGS) $(LDFLAGS) -o $@ $*.c

```

```

25026 .f~:
25027 $(GET) $(GFLAGS) -p $< > $*.f
25028 $(FC) $(FFLAGS) $(LDFLAGS) -o $@ $*.f

```

```

25029 .sh~:
25030 $(GET) $(GFLAGS) -p $< > $*.sh
25031 cp $*.sh $@
25032 chmod a+x $@
25033

```

#### 25034 *DOUBLE SUFFIX RULES*

```

25035 .c.o:
25036 $(CC) $(CFLAGS) -c $<

```

```

25037 .f.o:
25038 $(FC) $(FFLAGS) -c $<

```

```

25039 .y.o:
25040 $(YACC) $(YFLAGS) $<
25041 $(CC) $(CFLAGS) -c y.tab.c
25042 rm -f y.tab.c
25043 mv y.tab.o $@

```

```

25044 .l.o:
25045 $(LEX) $(LFLAGS) $<
25046 $(CC) $(CFLAGS) -c lex.yy.c
25047 rm -f lex.yy.c
25048 mv lex.yy.o $@

```

```

25049 .y.c:
25050 $(YACC) $(YFLAGS) $<
25051 mv y.tab.c $@

```

```

25052 .l.c:
25053 $(LEX) $(LFLAGS) $<

```

```

25054 mv lex.yy.c $@
25055 xsi .c~.o:
25056 $(GET) $(GFLAGS) -p $< > $*.c
25057 $(CC) $(CFLAGS) -c $*.c
25058
25059 .f~.o:
25059 $(GET) $(GFLAGS) -p $< > $*.f
25060 $(FC) $(FFLAGS) -c $*.f
25061
25062 .y~.o:
25062 $(GET) $(GFLAGS) -p $< > $*.y
25063 $(YACC) $(YFLAGS) $*.y
25064 $(CC) $(CFLAGS) -c y.tab.c
25065 rm -f y.tab.c
25066 mv y.tab.o $@
25067
25068 .l~.o:
25068 $(GET) $(GFLAGS) -p $< > $*.l
25069 $(LEX) $(LFLAGS) $*.l
25070 $(CC) $(CFLAGS) -c lex.yy.c
25071 rm -f lex.yy.c
25072 mv lex.yy.o $@
25073
25074 .y~.c:
25074 $(GET) $(GFLAGS) -p $< > $*.y
25075 $(YACC) $(YFLAGS) $*.y
25076 mv y.tab.c $@
25077
25078 .l~.c:
25078 $(GET) $(GFLAGS) -p $< > $*.l
25079 $(LEX) $(LFLAGS) $*.l
25080 mv lex.yy.c $@
25081
25082 .c.a:
25083 $(CC) -c $(CFLAGS) $<
25084 $(AR) $(ARFLAGS) $@ $*.o
25085 rm -f $*.o
25086
25087 .f.a:
25087 $(FC) -c $(FFLAGS) $<
25088 $(AR) $(ARFLAGS) $@ $*.o
25089 rm -f $*.o
25090 EXIT STATUS
25091 When the -q option is specified, the make utility shall exit with one of the following values:
25092 0 Successful completion.
25093 1 The target was not up-to-date.
25094 >1 An error occurred.
25095 When the -q option is not specified, the make utility shall exit with one of the following values:
25096 0 Successful completion.
25097 >0 An error occurred.

```

## 25098 CONSEQUENCES OF ERRORS

25099 Default.

## 25100 APPLICATION USAGE

25101 If there is a source file (such as *./source.c*) and there are two SCCS files corresponding to it  
 25102 (*./s.source.c* and *./SCCS/s.source.c*), on XSI-conformant systems *make* uses the SCCS file in the  
 25103 current directory. However, users are advised to use the underlying SCCS utilities (*admin*, *delta*,  
 25104 *get*, and so on) or the *sccs* utility for all source files in a given directory. If both forms are used for  
 25105 a given source file, future developers are very likely to be confused.

25106 It is incumbent upon portable makefiles to specify the **.POSIX** special target in order to  
 25107 guarantee that they are not affected by local extensions.

25108 The **-k** and **-S** options are both present so that the relationship between the command line, the  
 25109 **MAKEFLAGS** variable, and the makefile can be controlled precisely. If the **k** flag is passed in  
 25110 **MAKEFLAGS** and a command is of the form:

25111 \$(MAKE) -S foo

25112 then the default behavior is restored for the child *make*.

25113 When the **-n** option is specified, it is always added to **MAKEFLAGS**. This allows a recursive  
 25114 *make -n target* to be used to see all of the action that would be taken to update *target*.

25115 Because of widespread historical practice, interpreting a '#' number sign inside a variable as  
 25116 the start of a comment has the unfortunate side effect of making it impossible to place a number  
 25117 sign in a variable, thus forbidding something like:

25118 CFLAGS = "-D COMMENT\_CHAR='#'"

25119 Many historical *make* utilities stop chaining together inference rules when an intermediate target  
 25120 is nonexistent. For example, it might be possible for a *make* to determine that both *.y.c* and *.c.o*  
 25121 could be used to convert *.y* to a *.o*. Instead, in this case, *make* requires the use of a *.y.o* rule.

25122 The best way to provide portable makefiles is to include all of the rules needed in the makefile  
 25123 itself. The rules provided use only features provided by other parts of this volume of  
 25124 IEEE Std. 1003.1-200x. The default rules include rules for optional commands in this volume of  
 25125 IEEE Std. 1003.1-200x. Only rules pertaining to commands that are provided are needed in an  
 25126 implementation's default set.

25127 Macros used within other macros are evaluated when the new macro is used rather than when  
 25128 the new macro is defined. Therefore:

```
25129 MACRO = value1
25130 NEW = $(MACRO)
25131 MACRO = value2
```

```
25132 target:
25133 echo $(NEW)
```

25134 would produce *value2* and not *value1* since **NEW** was not expanded until it was needed in the  
 25135 *echo* command line.

25136 Some historical applications have been known to intermix *target\_name* and *macro=name* operands  
 25137 on the command line, expecting that all of the macros are processed before any of the targets are  
 25138 dealt with. Portable applications do not do this, although some backward compatibility support  
 25139 may be included in some implementations.

25140 The following characters in file names may give trouble: '=', ':', '\', '\'', and '@'. For  
 25141 inference rules, the description of \$< and \$? seem similar. However, an example shows the

25142 minor difference. In a makefile containing:  
 25143 `foo.o: foo.h`  
 25144 if **foo.h** is newer than **foo.o**, yet **foo.c** is older than **foo.o**, the built-in rule to make **foo.o** from  
 25145 **foo.c** is used, with `$<` equal to **foo.c** and `$?` equal to **foo.h**. If **foo.c** is also newer than **foo.o**, `$<` is  
 25146 equal to **foo.c** and `$?` is equal to **foo.h foo.c**.

#### 25147 EXAMPLES

25148 1. The following command:  
 25149 `make`  
 25150 makes the first target found in the makefile.

25151 2. The following command:  
 25152 `make junk`  
 25153 makes the target **junk**.

25154 3. The following makefile says that **pgm** depends on two files, **a.o** and **b.o**, and that they in  
 25155 turn depend on their corresponding source files (**a.c** and **b.c**), and a common file **incl.h**:

```
25156 pgm: a.o b.o
25157 c89 a.o b.o -o pgm
25158 a.o: incl.h a.c
25159 c89 -c a.c
25160 b.o: incl.h b.c
25161 c89 -c b.c
```

25162 4. An example for making optimized **.o** files from **.c** files is:

```
25163 .c.o:
25164 c89 -c -O $*.c
```

25165 or:

```
25166 .c.o:
25167 c89 -c -O $<
```

25168 5. The most common use of the archive interface follows. Here, it is assumed that the source  
 25169 files are all C-language source:

```
25170 lib: lib(file1.o) lib(file2.o) lib(file3.o)
25171 @echo lib is now up-to-date
```

25172 The **.c.a** rule is used to make **file1.o**, **file2.o**, and **file3.o** and insert them into **lib**.

25173 The treatment of escaped <newline> characters throughout the makefile is historical  
 25174 practice. For example, the inference rule:

```
25175 .c.o\
25176 :
```

25177 works, and the macro:

```
25178 f= bar baz\
25179 biz
25180 a:
25181 echo ==$f==
```

```

25182 echoes "==bar baz biz==".
25183 If $? were:
25184 /usr/include/stdio.h /usr/include/unistd.h foo.h
25185 then $(?D) would be:
25186 /usr/include /usr/include .
25187 and $(?F) would be:
25188 stdio.h unistd.h foo.h
25189 6. The contents of the built-in rules can be viewed by running:
25190 make -p -f /dev/null 2>/dev/null

```

#### 25191 RATIONALE

25192 The *make* utility described in this volume of IEEE Std. 1003.1-200x is intended to provide the  
 25193 means for changing portable source code into executables that can be run on a  
 25194 IEEE Std. 1003.1-200x-conforming system. It reflects the most common features present in  
 25195 System V and BSD *makes*.

25196 Historically, the *make* utility has been an especially fertile ground for vendor and research  
 25197 organization-specific syntax modifications and extensions. Examples include:

- 25198 • Syntax supporting parallel execution (such as from various multiprocessor vendors, GNU,  
 25199 and others)
- 25200 • Additional “operators” separating targets and their prerequisites (System V, BSD, and  
 25201 others)
- 25202 • Specifying that command lines containing the strings `$(MAKE)` and `$(MAKE)` are executed  
 25203 when the `-n` option is specified (GNU and System V)
- 25204 • Modifications of the meaning of internal macros when referencing libraries (BSD and others)
- 25205 • Using a single instance of the shell for all of the command lines of the target (BSD and others)
- 25206 • Allowing spaces as well as tabs to delimit command lines (BSD)
- 25207 • Adding C preprocessor-style “include” and “ifdef” constructs (System V, GNU, BSD, and  
 25208 others)
- 25209 • Remote execution of command lines (Sprite and others)
- 25210 • Specifying additional special targets (BSD, System V, and most others)

25211 Additionally, many vendors and research organizations have rethought the basic concepts of  
 25212 *make*, creating vastly extended, as well as completely new, syntaxes. Each of these versions of  
 25213 *make* fulfills the needs of a different community of users; it is unreasonable for this volume of  
 25214 IEEE Std. 1003.1-200x to require behavior that would be incompatible (and probably inferior) to  
 25215 historical practice for such a community.

25216 In similar circumstances, when the industry has enough sufficiently incompatible formats as to  
 25217 make them irreconcilable, this volume of IEEE Std. 1003.1-200x has followed one or both of two  
 25218 courses of action. Commands have been renamed (*cksum*, *echo*, and *pax*) and/or command line  
 25219 options have been provided to select the desired behavior (*grep*, *od*, and *pax*).

25220 Because the syntax specified for the *make* utility is, by and large, a subset of the syntaxes  
 25221 accepted by almost all versions of *make*, it was decided that it would be counter-productive to  
 25222 change the name. And since the makefile itself is a basic unit of portability, it would not be

25223 completely effective to reserve a new option letter, such as *make -P*, to achieve the portable  
25224 behavior. Therefore, the special target **.POSIX** was added to the makefile, allowing users to  
25225 specify “standard” behavior. This special target does not preclude extensions in the *make* utility,  
25226 nor does it preclude such extensions being used by the makefile specifying the target; it does,  
25227 however, preclude any extensions from being applied that could alter the behavior of previously  
25228 valid syntax; such extensions must be controlled via command line options or new special  
25229 targets. It is incumbent upon portable makefiles to specify the **.POSIX** special target in order to  
25230 guarantee that they are not affected by local extensions.

25231 The portable version of *make* described in this reference page is not intended to be the state-of-  
25232 the-art software generation tool and, as such, some newer and more leading-edge features have  
25233 not been included. An attempt has been made to describe the portable makefile in a manner that  
25234 does not preclude such extensions as long as they do not disturb the portable behavior described  
25235 here.

25236 When the **-n** option is specified, it is always added to **MAKEFLAGS**. This allows a recursive  
25237 *make -n target* to be used to see all of the action that would be taken to update *target*.

25238 The definition of **MAKEFLAGS** allows both the System V letter string and the BSD command line  
25239 formats. The two formats are sufficiently different to allow implementations to support both  
25240 without ambiguity.

25241 Early proposals stated that an “unquoted” number sign was treated as the start of a comment.  
25242 The *make* utility does not pay any attention to quotes. A number sign starts a comment  
25243 regardless of its surroundings.

25244 The text about “other implementation-dependent path names may also be tried” in addition to  
25245 **./makefile** and **./Makefile** is to allow such extensions as **SCCS/s.Makefile** and other variations.  
25246 It was made an implementation-dependent requirement (as opposed to unspecified behavior) to  
25247 highlight surprising implementations that might select something unexpected like  
25248 **/etc/Makefile**.

25249 Early proposals contained the macro **NPROC** as a means of specifying that *make* should use *n*  
25250 processes to do the work required. While this feature is a valuable extension for many systems, it  
25251 is not common usage and could require other non-trivial extensions to makefile syntax. This  
25252 extension is not required by this volume of IEEE Std. 1003.1-200x, but could be provided as a  
25253 compatible extension. The macro **PARALLEL** is used by some historical systems with essentially  
25254 the same meaning (but without using a name that is a common system limit value). It is  
25255 suggested that implementors recognize the existing use of **NPROC** and/or **PARALLEL** as  
25256 extensions to *make*.

25257 The default rules are based on System V. The default **CC=** value is *c89* instead of *cc* because this  
25258 volume of IEEE Std. 1003.1-200x does not standardize the utility named *cc*. Thus, every  
25259 conforming application would be required to define **CC=c89** to expect to run. There is no  
25260 advantage conferred by the hope that the makefile might hit the “preferred” compiler because  
25261 this cannot be guaranteed to work. Also, since the portable makescript can only use the *c89*  
25262 options, no advantage is conferred in terms of what the script can do. It is a quality-of-  
25263 implementation issue as to whether *c89* is as valuable as *cc*.

25264 The **-d** option to *make* is frequently used to produce debugging information, but is too  
25265 implementation-dependent to add to this volume of IEEE Std. 1003.1-200x.

25266 The **-p** option is not passed in **MAKEFLAGS** on most historical implementations and to change  
25267 this would cause many implementations to break without sufficiently increased portability.

25268 Commands that begin with a plus sign (**' + '**) are executed even if the **-n** option is present. Based  
25269 on the GNU version of *make*, the behavior of **-n** when the plus-sign prefix is encountered has

been extended to apply to `-q` and `-t` as well. However, the System V convention of forcing command execution with `-n` when the command line of a target contains either of the strings `$(MAKE)` or `${MAKE}` has not been adopted. This functionality appeared in early proposals, but the danger of this approach was pointed out with the following example of a portion of a makefile:

```
subdir:
 cd subdir; rm all_the_files; $(MAKE)
```

The loss of the System V behavior in this case is well-balanced by the safety afforded to other makefiles that were not aware of this situation. In any event, the command line plus-sign prefix can provide the desired functionality.

The double colon in the target rule format is supported in BSD systems to allow more than one target line containing the same target name to have commands associated with it. Since this is not functionality described in the SVID or XPG3 it has been allowed as an extension, but not mandated.

The default rules are provided with text specifying that the built-in rules shall be the same *as if* the listed set were used. The intent is that implementations should be able to use the rules without change, but will be allowed to alter them in ways that do not affect the primary behavior.

The best way to provide portable makefiles is to include all of the rules needed in the makefile itself. The rules provided use only features provided by other portions of this volume of IEEE Std. 1003.1-200x. The default rules include rules for optional commands in this volume of IEEE Std. 1003.1-200x. Only rules pertaining to commands that are provided are needed in the default set of an implementation.

One point of discussion was whether to drop the default rules list from this volume of IEEE Std. 1003.1-200x. They provide convenience, but do not enhance portability of applications. The prime benefit is in portability of users who wish to type *make command* and have the command build from a **command.c** file.

The historical *MAKESHELL* feature was omitted. In some implementations it is used to let a user override the shell to be used to run *make* commands. This was confusing; for a portable *make*, the shell should be chosen by the makefile writer or specified on the *make* command line and not by a user running *make*.

The *make* utilities in most historical implementations process the prerequisites of a target in left-to-right order, and the makefile format requires this. It supports the standard idiom used in many makefiles that produce yacc programs; for example:

```
foo: y.tab.o lex.o main.o
 $(CC) $(CFLAGS) -o $@ t.tab.o lex.o main.o
```

In this example, if *make* chose any arbitrary order, the **lex.o** might not be made with the correct **y.tab.h**. Although there may be better ways to express this relationship, it is widely used historically. Implementations that desire to update prerequisites in parallel should require an explicit extension to *make* or the makefile format to accomplish it, as described previously.

The algorithm for determining a new entry for target rules is partially unspecified. Some historical *makes* allow blank, empty, or comment lines within the collection of commands marked by leading `<tab>s`. A conforming makefile must ensure that each command starts with a `<tab>`, but implementations are free to ignore blank, empty, and comment lines without triggering the start of a new entry.



The ASYNCHRONOUS EVENTS section includes having SIGTERM and SIGHUP, along with the more traditional SIGINT and SIGQUIT, remove the current target unless directed not to do so. SIGTERM and SIGHUP were added to parallel other utilities that have historically cleaned up their work as a result of these signals. When *make* receives any signal other than SIGQUIT, it is required to resend itself the signal it received so that it exits with a status that reflects the signal. The results from SIGQUIT are partially unspecified because, on systems that create **core** files upon receipt of SIGQUIT, the **core** from *make* would conflict with a core file from the command that was running when the SIGQUIT arrived. The main concern was to prevent damaged files from appearing up-to-date when *make* is rerun.

The **.PRECIOUS** special target was extended to affect all targets globally (by specifying no prerequisites). The **.IGNORE** and **.SILENT** special targets were extended to allow prerequisites; it was judged to be more useful in some cases to be able to turn off errors or echoing for a list of targets than for the entire makefile. These extensions to the *make* in System V were made to match historical practice from the BSD *make*.

Macros are not exported to the environment of commands to be run. This was never the case in any historical *make* and would have serious consequences. The environment is the same as the environment to *make* except that **MAKEFLAGS** and macros defined on the *make* command line are added.

Some implementations do not use *system()* for all command lines, as required by the portable makefile format; as a performance enhancement, they select lines without shell metacharacters for direct execution by *execve()*. There is no requirement that *system()* be used specifically, but merely that the same results be achieved. The metacharacters typically used to bypass the direct *execve()* execution have been any of:

```
= | ^ () ; & < > * ? [] : $ \ ' " \ \n
```

The default in some advanced versions of *make* is to group all the command lines for a target and execute them using a single shell invocation; the System V method is to pass each line individually to a separate shell. The single-shell method has the advantages in performance and the lack of a requirement for many continued lines. However, converting to this newer method has caused portability problems with many historical makefiles, so the behavior with the POSIX makefile is specified to be the same as that of System V. It is suggested that the special target **.ONESHELL** be used as an implementation extension to achieve the single-shell grouping for a target or group of targets.

Novice users of *make* have had difficulty with the historical need to start commands with a <tab> character. Since it is often difficult to discern differences between <tab> and <space> characters on terminals or printed listings, confusing bugs can arise. In early proposals, an attempt was made to correct this problem by allowing leading <blank>s instead of <tab>s. However, implementors reported many makefiles that failed in subtle ways following this change, and it is difficult to implement a *make* that unambiguously can differentiate between macro and command lines. There is extensive historical practice of allowing leading spaces before macro definitions. Forcing macro lines into column 1 would be a significant backwards-compatibility problem for some makefiles. Therefore, historical practice was restored.

The System V **INCLUDE** feature was considered, but not included. This would treat a line that began in the first column and contained **INCLUDE <filename>** as an indication to read <filename> at that point in the makefile. This is difficult to use in a portable way, and it raises concerns about nesting levels and diagnostics. System V, BSD, GNU, and others have used different methods for including files.

The System V dynamic dependency feature was not included. It would support:

25362        `cat: $$@.c`

25363        that would expand to;

25364        `cat: cat.c`

25365        This feature exists only in the new version of System V *make* and, while useful, is not in wide  
25366        usage. This means that macros are expanded twice for prerequisites: once at makefile parse time  
25367        and once at target update time.

25368        Consideration was given to adding metarules to the POSIX *make*. This would make `%o: %c` the  
25369        same as `.c.o:`. This is quite useful and available from some vendors, but it would cause too many  
25370        changes to this *make* to support. It would have introduced rule chaining and new substitution  
25371        rules. However, the rules for target names have been set to reserve the `'%'` and `' '` characters.  
25372        These are traditionally used to implement metarules and quoting of target names, respectively.  
25373        Implementors are strongly encouraged to use these characters only for these purposes.

25374        A request was made to extend the suffix delimiter character from a period to any character. The  
25375        metarules feature in newer *makes* solves this problem in a more general way. This volume of  
25376        IEEE Std. 1003.1-200x is staying with the more conservative historical definition.

25377        The standard output format for the `-p` option is not described because it is primarily a  
25378        debugging option and because the format is not generally useful to programs. In historical  
25379        implementations the output is not suitable for use in generating makefiles. The `-p` format has  
25380        been variable across historical implementations. Therefore, the definition of `-p` was only to  
25381        provide a consistently named option for obtaining *make* script debugging information.

25382        Some historical implementations have not cleared the suffix list with `-r`.

25383        Implementations should be aware that some historical applications have intermixed *target\_name*  
25384        and *macro=value* operands on the command line, expecting that all of the macros are processed  
25385        before any of the targets are dealt with. Portable applications do not do this, but some  
25386        backwards-compatibility support may be warranted.

25387        Empty inference rules are specified with a semicolon command rather than omitting all  
25388        commands, as described in an early proposal. The latter case has no traditional meaning and is  
25389        reserved for implementation extensions, such as in GNU *make*.

## 25390 **FUTURE DIRECTIONS**

25391        None.

## 25392 **SEE ALSO**

25393        *ar*, *c89*, *get*, *lex*, *sh*, *yacc*, the System Interfaces volume of IEEE Std. 1003.1-200x, *system()*

## 25394 **CHANGE HISTORY**

25395        First released in Issue 2.

## 25396 **Issue 4**

25397        Aligned with the ISO/IEC 9945-2: 1993 standard.

## 25398 **Issue 4, Version 2**

25399        Under **Default Rules**, the string `"-G$@"` is deleted from the line referencing *sccs*.

## 25400 **Issue 5**

25401        FUTURE DIRECTIONS section added.

## 25402 **Issue 6**

25403        This utility is now marked as part of the Software Development Utilities option.

25404        The Open Group corrigenda item U029/1 has been applied, correcting a typographical error in  
25405        the SPECIAL TARGETS section.

|       |                                                                                                        |  |
|-------|--------------------------------------------------------------------------------------------------------|--|
| 25406 | In the ENVIRONMENT VARIABLES section, the <i>PROJECTDIR</i> description is updated from                |  |
| 25407 | “otherwise, the home directory of a user of that name is examined” to “otherwise, the value of         |  |
| 25408 | <i>PROJECTDIR</i> is treated as a user name and that user’s initial working directory is examined”.    |  |
| 25409 | It is specified whether the command line is related to the makefile or to the <i>make</i> command, and |  |
| 25410 | the macro processing rules are updated to align with the IEEE P1003.2b draft standard.                 |  |
| 25411 | The normative text is reworded to avoid use of the term “must” for application requirements.           |  |

25412 **NAME**

25413       man — display system documentation

25414 **SYNOPSIS**25415       man [-k] *name*...25416 **DESCRIPTION**

25417       The *man* utility shall write information about each of the *name* operands. If *name* is the name of a  
 25418       standard utility, *man* at a minimum shall write a message describing the syntax used by the  
 25419       standard utility, its options, and operands. If more information is available, the *man* utility shall  
 25420       provide it in an implementation-dependent manner.

25421       An implementation may provide information for values of *name* other than the standard utilities.  
 25422       Standard utilities that are listed as optional and that are not supported by the implementation  
 25423       either shall cause a brief message indicating that fact to be displayed or shall cause a full display  
 25424       of information as described previously.

25425 **OPTIONS**

25426       The *man* utility shall conform to the System Interface Definitions volume of  
 25427       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

25428       The following option shall be supported:

25429       **-k**       Interpret *name* operands as keywords to be used in searching a utilities summary  
 25430       database that contains a brief purpose entry for each standard utility and write lines  
 25431       from the summary database that match any of the keywords. The keyword search shall  
 25432       produce results that are the equivalent of the output of the following command:

```
25433 grep -Ei '
25434 name
25435 name
25436 ...
25437 ' summary-database
```

25438       This assumes that the *summary-database* is a text file with a single entry per line; this  
 25439       organization is not required and the example using *grep -Ei* is merely illustrative of the  
 25440       type of search intended. The purpose entry to be included in the database shall consist  
 25441       of a terse description of the purpose of the utility.

25442 **OPERANDS**

25443       The following operand shall be supported:

25444       *name*       A keyword or the name of a standard utility. When **-k** is not specified and *name*  
 25445       does not represent one of the standard utilities, the results are unspecified.

25446 **STDIN**

25447       Not used.

25448 **INPUT FILES**

25449       None.

25450 **ENVIRONMENT VARIABLES**25451       The following environment variables shall affect the execution of *man*:

25452       **LANG**       Provide a default value for the internationalization variables that are unset or null.  
 25453       If *LANG* is unset or null, the corresponding value from the implementation-  
 25454       dependent default locale shall be used. If any of the internationalization variables  
 25455       contains an invalid setting, the utility shall behave as if none of the variables had  
 25456       been defined.

25457 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 25458 internationalization variables.

25459 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 25460 characters (for example, single-byte as opposed to multi-byte characters in  
 25461 arguments and in the summary database). The value of **LC\_CTYPE** need not affect  
 25462 the format of the information written about the name operands.

25463 **LC\_MESSAGES**  
 25464 Determine the locale that should be used to affect the format and contents of  
 25465 diagnostic messages written to standard error and informative messages written to  
 25466 standard output.

25467 XSI **NLSPATH** Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

25468 **PAGER** Determine an output filtering command for writing the output to a terminal. Any  
 25469 string acceptable as a *command\_string* operand to the **sh -c** command shall be valid.  
 25470 When standard output is a terminal device, the reference page output shall be  
 25471 piped through the command. If the **PAGER** variable is null or not set, the  
 25472 command shall be either *more* or another paginator utility documented in the  
 25473 system documentation.

25474 **ASYNCHRONOUS EVENTS**

25475 Default.

25476 **STDOUT**  
 25477 The *man* utility shall write text describing the syntax of the utility *name*, its options and its  
 25478 operands, or, when **-k** is specified, lines from the summary database. The format of this text is  
 25479 implementation-dependent.

25480 **STDERR**  
 25481 Used only for diagnostic messages.

25482 **OUTPUT FILES**  
 25483 None.

25484 **EXTENDED DESCRIPTION**  
 25485 None.

25486 **EXIT STATUS**  
 25487 The following exit values shall be returned:

25488 0 Successful completion.

25489 >0 An error occurred.

25490 **CONSEQUENCES OF ERRORS**  
 25491 Default.

25492 **APPLICATION USAGE**  
 25493 None.

25494 **EXAMPLES**  
 25495 None.

25496 **RATIONALE**  
 25497 It is recognized that the *man* utility is only of minimal usefulness as specified. The opinion of the  
 25498 standard developers was strongly divided as to how much or how little information *man* should  
 25499 be required to provide. They considered, however, that the provision of some portable way of  
 25500 accessing documentation would aid user portability. The arguments against a fuller

25501 specification were:

- 25502 • Large quantities of documentation should not be required on a system that does not have
- 25503 excess disk space.
- 25504 • The current manual system does not present information in a manner that greatly aids user
- 25505 portability.
- 25506 • A “better help system” is currently an area in which vendors feel that they can add value to
- 25507 their POSIX implementations.

25508 The `-f` option was considered, but due to implementation differences, it was not included in this

25509 volume of IEEE Std. 1003.1-200x.

25510 The description was changed to be more specific about what has to be displayed for a utility.

25511 The standard developers considered it insufficient to allow a display of only the synopsis

25512 without giving a short description of what each option and operand does.

25513 The “purpose” entry to be included in the database can be similar to the section title (less the

25514 numeric prefix) from this volume of IEEE Std. 1003.1-200x for each utility. These titles are

25515 similar to those used in historical systems for this purpose.

25516 See *mailx* for rationale concerning the default paginator.

25517 The caveat in the *LC\_CTYPE* description was added because it is not a requirement that an

25518 implementation provide reference pages for all of its supported locales on each system;

25519 changing *LC\_CTYPE* does not necessarily translate the reference page into another language.

25520 This is equivalent to the current state of *LC\_MESSAGES* in IEEE Std. 1003.1-200x—locale-specific

25521 messages are not yet a requirement.

25522 The historical *MANPATH* variable is not included in POSIX because no attempt is made to

25523 specify naming conventions for reference page files, nor even to mandate that they are files at

25524 all. In some systems they could be a true database, a hypertext file, or even fixed strings within

25525 the *man* executable. The standard developers considered the portability of reference pages to be

25526 outside their scope of work (and more appropriate to the POSIX.7 working group developing

25527 application-installation tools). However, users should be aware that *MANPATH* is implemented

25528 on a number of historical systems and that it can be used to tailor the search pattern for reference

25529 pages from the various categories (utilities, functions, file formats, and so on) when the system

25530 administrator reveals the location and conventions for reference pages on the system.

25531 The keyword search can rely on at least the text of the section titles from these utility

25532 descriptions, and the implementation may add more keywords. The term “section titles” refers

25533 to the strings such as:

25534 `man` — Display system documentation

25535 `ps` — Report process status

25536 **FUTURE DIRECTIONS**

25537 None.

25538 **SEE ALSO**

25539 *more*

25540 **CHANGE HISTORY**

25541 First released in Issue 4.

25542 **Issue 5**

25543 FUTURE DIRECTIONS section added.

25544 **NAME**

25545           mesg — permit or deny messages

25546 **SYNOPSIS**25547 UP       mesg [*y*|*n*]

25548

25549 **DESCRIPTION**

25550       The *mesg* utility shall control whether other users are allowed to send messages via *write*, *talk*, or  
 25551       other utilities to a terminal device. The terminal device affected shall be determined by searching  
 25552       for the first terminal in the sequence of devices associated with standard input, standard output,  
 25553       and standard error, respectively. With no arguments, *mesg* shall report the current state without  
 25554       changing it. Processes with appropriate privileges may be able to send messages to the terminal  
 25555       independent of the current state.

25556 **OPTIONS**

25557       None.

25558 **OPERANDS**

25559       The following operands shall be supported in the POSIX locale:

25560       *y*           Grant permission to other users to send messages to the terminal device.25561       *n*           Deny permission to other users to send messages to the terminal device.25562 **STDIN**

25563       Not used.

25564 **INPUT FILES**

25565       None.

25566 **ENVIRONMENT VARIABLES**25567       The following environment variables shall affect the execution of *mesg*:

25568       *LANG*       Provide a default value for the internationalization variables that are unset or null.  
 25569       If *LANG* is unset or null, the corresponding value from the implementation-  
 25570       dependent default locale shall be used. If any of the internationalization variables  
 25571       contains an invalid setting, the utility shall behave as if none of the variables had  
 25572       been defined.

25573       *LC\_ALL*      If set to a non-empty string value, override the values of all the other  
 25574       internationalization variables.

25575       *LC\_CTYPE*   Determine the locale for the interpretation of sequences of bytes of text data as  
 25576       characters (for example, single-byte as opposed to multi-byte characters in  
 25577       arguments).

25578       *LC\_MESSAGES*

25579       Determine the locale that should be used to affect the format and contents of  
 25580       diagnostic messages written (by *mesg*) to standard error.

25581 XSI       *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.25582 **ASYNCHRONOUS EVENTS**

25583       Default.

25584 **STDOUT**25585       If no operand is specified, *mesg* shall display the current terminal state in an unspecified format.



25586 **STDERR**

25587           Used only for diagnostic messages.

25588 **OUTPUT FILES**

25589           None.

25590 **EXTENDED DESCRIPTION**

25591           None.

25592 **EXIT STATUS**

25593           The following exit values shall be returned:

25594           0   Receiving messages is allowed.

25595           1   Receiving messages is not allowed.

25596           >1  An error occurred.

25597 **CONSEQUENCES OF ERRORS**

25598           Default.

25599 **APPLICATION USAGE**

25600           The mechanism by which the message status of the terminal is changed is unspecified.  
25601           Therefore, unspecified actions may cause the status of the terminal to change after *mesg* has  
25602           successfully completed. These actions may include, but are not limited to: another invocation of  
25603           the *mesg* utility, login procedures; invocation of the *stty* utility, invocation of the *chmod* utility or  
25604           *chmod()* function, and so on.

25605           Application writers should note that this utility need not be provided on systems that do not  
25606           support the User Portability Utilities option.

25607 **EXAMPLES**

25608           None.

25609 **RATIONALE**

25610           The terminal changed by *mesg* is that associated with the standard input, output, or error, rather  
25611           than the controlling terminal for the session. This is because users logged in more than once  
25612           should be able to change any of their login terminals without having to stop the job running in  
25613           those sessions. This is not a security problem involving the terminals of other users because  
25614           appropriate privileges would be required to affect the terminal of another user.

25615           The method of checking each of the first three file descriptors in sequence until a terminal is  
25616           found was adopted from System V.

25617           The file */dev/tty* is not specified for the terminal device because it was thought to be too  
25618           restrictive. Typical environment changes for the *n* operand are that write permissions are  
25619           removed for *others* and *group* from the appropriate device. It was decided to leave the actual  
25620           description of what is done as unspecified because of potential differences between  
25621           implementations.

25622           The format for standard output is unspecified because of differences between historical  
25623           implementations. This output is generally not useful to shell scripts (they can use the exit  
25624           status), so exact parsing of the output is unnecessary.

25625 **FUTURE DIRECTIONS**

25626           None.

25627 **SEE ALSO**25628 *talk, write*25629 **CHANGE HISTORY**

25630 First released in Issue 2.

25631 **Issue 4**

25632 Aligned with the ISO/IEC 9945-2: 1993 standard.

25633 **Issue 6**

25634 This utility is now marked as part of the User Portability Utilities option.

25635 **NAME**

25636           mkdir — make directories

25637 **SYNOPSIS**25638           mkdir [-p][-m *mode*] *dir*...25639 **DESCRIPTION**25640           The *mkdir* utility shall create the directories specified by the operands, in the order specified.

25641           For each *dir* operand, the *mkdir* utility shall perform actions equivalent to the *mkdir()* function  
 25642           defined in the System Interfaces volume of IEEE Std. 1003.1-200x, called with the following  
 25643           arguments:

- 25644           1. The *dir* operand is used as the *path* argument.
- 25645           2. The value of the bitwise-inclusive OR of S\_IRWXU, S\_IRWXG, and S\_IRWXO is used as  
 25646           the *mode* argument. (If the **-m** option is specified, the *mode* option-argument overrides this  
 25647           default.)

25648 **OPTIONS**

25649           The *mkdir* utility shall conform to the System Interface Definitions volume of  
 25650           IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

25651           The following options shall be supported:

25652           **-m *mode***     Set the file permission bits of the newly-created directory to the specified *mode*  
 25653                           value. The *mode* option-argument shall be the same as the *mode* operand defined  
 25654                           for the *chmod* utility. In the *symbolic\_mode* strings, the *op* characters '+' and '-'  
 25655                           shall be interpreted relative to an assumed initial mode of *a=rwx*; '+' shall add  
 25656                           permissions to the default mode, '-' shall delete permissions from the default  
 25657                           mode.

25658           **-p**           Create any missing intermediate path name components.  
 25659                           For each *dir* operand that does not name an existing directory, effects equivalent to  
 25660                           those caused by the following command shall occur:

```
25661 mkdir -p -m $(umask -S),u+wx $(dirname dir) &&
25662 mkdir [-m mode] dir
```

25663                           where the **-m *mode*** option represents that option supplied to the original  
 25664                           invocation of *mkdir*, if any.

25665           Each *dir* operand that names an existing directory shall be ignored without error.25666 **OPERANDS**

25667           The following operand shall be supported:

25668           *dir*           A path name of a directory to be created.25669 **STDIN**

25670           Not used.

25671 **INPUT FILES**

25672           None.

25673 **ENVIRONMENT VARIABLES**25674           The following environment variables shall affect the execution of *mkdir*:

25675           **LANG**         Provide a default value for the internationalization variables that are unset or null.  
 25676                           If *LANG* is unset or null, the corresponding value from the implementation-  
 25677                           dependent default locale shall be used. If any of the internationalization variables

25678 contains an invalid setting, the utility shall behave as if none of the variables had  
 25679 been defined.

25680 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 25681 internationalization variables.

25682 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 25683 characters (for example, single-byte as opposed to multi-byte characters in  
 25684 arguments).

25685 **LC\_MESSAGES**  
 25686 Determine the locale that should be used to affect the format and contents of  
 25687 diagnostic messages written to standard error.

25688 XSI **NLSPATH** Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

25689 **ASYNCHRONOUS EVENTS**  
 25690 Default.

25691 **STDOUT**  
 25692 Not used.

25693 **STDERR**  
 25694 Used only for diagnostic messages.

25695 **OUTPUT FILES**  
 25696 None.

25697 **EXTENDED DESCRIPTION**  
 25698 None.

25699 **EXIT STATUS**  
 25700 The following exit values shall be returned:

25701 0 All the specified directories were created successfully or the **-p** option was specified and all  
 25702 the specified directories now exist.

25703 >0 An error occurred.

25704 **CONSEQUENCES OF ERRORS**  
 25705 Default.

25706 **APPLICATION USAGE**  
 25707 The default file mode for directories is *a=rwx* (777 on most systems) with selected permissions  
 25708 removed in accordance with the file mode creation mask. For intermediate path name  
 25709 components created by *mkdir*, the mode is the default modified by *u+wx* so that the  
 25710 subdirectories can always be created regardless of the file mode creation mask; if different  
 25711 ultimate permissions are desired for the intermediate directories, they can be changed  
 25712 afterwards with *chmod*.

25713 Note that some of the requested directories may have been created even if an error occurs.

25714 **EXAMPLES**  
 25715 None.

25716 **RATIONALE**  
 25717 The System V **-m** option was included to control the file mode.

25718 The System V **-p** option was included to create any needed intermediate directories and to  
 25719 complement the functionality provided by *rmdir* for removing directories in the path prefix as  
 25720 they become empty. Because no error is produced if any path component already exists, the **-p**

- 25721 option is also useful to ensure that a particular directory exists.
- 25722 The functionality of *mkdir* is described substantially through a reference to the *mkdir()* function
- 25723 in the System Interfaces volume of IEEE Std. 1003.1-200x. For example, by default, the mode of
- 25724 the directory is affected by the file mode creation mask in accordance with the specified
- 25725 behavior of the *mkdir()* function. In this way, there is less duplication of effort required for
- 25726 describing details of the directory creation.
- 25727 **FUTURE DIRECTIONS**
- 25728 None.
- 25729 **SEE ALSO**
- 25730 *rm*, *rmdir*, *umask*, the System Interfaces volume of IEEE Std. 1003.1-200x, *mkdir()*
- 25731 **CHANGE HISTORY**
- 25732 First released in Issue 2.
- 25733 **Issue 4**
- 25734 Aligned with the ISO/IEC 9945-2:1993 standard.
- 25735 **Issue 5**
- 25736 FUTURE DIRECTIONS section added.

25737 **NAME**

25738 mkfifo — make FIFO special files

25739 **SYNOPSIS**25740 mkfifo [-m *mode*] *file*...25741 **DESCRIPTION**25742 The *mkfifo* utility shall create the FIFO special files specified by the operands, in the order  
25743 specified.25744 For each *file* operand, the *mkfifo* utility shall perform actions equivalent to the *mkfifo()* function  
25745 defined in the System Interfaces volume of IEEE Std. 1003.1-200x, called with the following  
25746 arguments:

- 25747 1. The *file* operand is used as the *path* argument.
- 25748 2. The value of the bitwise-inclusive OR of S\_IRUSR, S\_IWUSR, S\_IRGRP, S\_IWGRP,  
25749 S\_IROTH, and S\_IWOTH is used as the *mode* argument. (If the -m option is specified, the  
25750 value of the *mkfifo()* *mode* argument is unspecified, but the FIFO shall at no time have  
25751 permissions less restrictive than the -m *mode* option-argument.)

25752 **OPTIONS**25753 The *mkfifo* utility shall conform to the System Interface Definitions volume of  
25754 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

25755 The following option shall be supported:

25756 -m *mode* Set the file permission bits of the newly-created FIFO to the specified *mode* value.  
 25757 The *mode* option-argument shall be the same as the *mode* operand defined for the  
 25758 *chmod* utility. In the *symbolic\_mode* strings, the *op* characters '+' and '-' shall be  
 25759 interpreted relative to an assumed initial mode of *a=rw*.

25760 **OPERANDS**

25761 The following operand shall be supported:

25762 *file* A path name of the FIFO special file to be created.25763 **STDIN**

25764 Not used.

25765 **INPUT FILES**

25766 None.

25767 **ENVIRONMENT VARIABLES**25768 The following environment variables shall affect the execution of *mkfifo*:

25769 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 25770 If *LANG* is unset or null, the corresponding value from the implementation-  
 25771 dependent default locale shall be used. If any of the internationalization variables  
 25772 contains an invalid setting, the utility shall behave as if none of the variables had  
 25773 been defined.

25774 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 25775 internationalization variables.

25776 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 25777 characters (for example, single-byte as opposed to multi-byte characters in  
 25778 arguments).

25779 *LC\_MESSAGES*

25780 Determine the locale that should be used to affect the format and contents of

- 25781 diagnostic messages written to standard error.
- 25782 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.
- 25783 **ASYNCHRONOUS EVENTS**
- 25784 Default.
- 25785 **STDOUT**
- 25786 Not used.
- 25787 **STDERR**
- 25788 Used only for diagnostic messages.
- 25789 **OUTPUT FILES**
- 25790 None.
- 25791 **EXTENDED DESCRIPTION**
- 25792 None.
- 25793 **EXIT STATUS**
- 25794 The following exit values shall be returned:
- 25795 0 All the specified FIFO special files were created successfully.
- 25796 >0 An error occurred.
- 25797 **CONSEQUENCES OF ERRORS**
- 25798 Default.
- 25799 **APPLICATION USAGE**
- 25800 None.
- 25801 **EXAMPLES**
- 25802 None.
- 25803 **RATIONALE**
- 25804 This new utility was added to permit shell applications to create FIFO special files.
- 25805 The **-m** option was added to control the file mode, for consistency with the similar functionality
- 25806 provided the *mkdir* utility.
- 25807 Early proposals included a **-p** option similar to the *mkdir -p* option that created intermediate
- 25808 directories leading up to the FIFO specified by the final component. This was removed because
- 25809 it is not commonly needed and is not common practice with similar utilities.
- 25810 The functionality of *mkfifo* is described substantially through a reference to the *mkfifo()* function
- 25811 in the System Interfaces volume of IEEE Std. 1003.1-200x. For example, by default, the mode of
- 25812 the FIFO file is affected by the file mode creation mask in accordance with the specified behavior
- 25813 of the *mkfifo()* function. In this way, there is less duplication of effort required for describing
- 25814 details of the file creation.
- 25815 **FUTURE DIRECTIONS**
- 25816 None.
- 25817 **SEE ALSO**
- 25818 *umask*, the System Interfaces volume of IEEE Std. 1003.1-200x, *mkfifo()*
- 25819 **CHANGE HISTORY**
- 25820 First released in Issue 3.

25821 **Issue 4**  
25822       Aligned with the ISO/IEC 9945-2: 1993 standard.



25823 **NAME**25824        *more* — display files on a page-by-page basis25825 **SYNOPSIS**25826 UP        *more* [-ceisu][-n *number*][-p *command*][-t *tagstring*][*file* ...]

25827

25828 **DESCRIPTION**

25829        The *more* utility shall read files and either write them to the terminal on a page-by-page basis or  
 25830        filter them to standard output. If standard output is not a terminal device, all input files shall be  
 25831        copied to standard output in their entirety, without modification, except as specified for the **-s**  
 25832        option. If standard output is a terminal device, the files shall be written a number of lines (one  
 25833        screenful) at a time under the control of user commands. See the EXTENDED DESCRIPTION  
 25834        section.

25835        Certain block-mode terminals do not have all the capabilities necessary to support the complete  
 25836        *more* definition; they are incapable of accepting commands that are not terminated with a  
 25837        <newline> character. Implementations that support such terminals shall provide an operating  
 25838        mode to *more* in which all commands can be terminated with a <newline> character on those  
 25839        terminals. This mode:

- 25840        • Shall be documented in the system documentation
- 25841        • Shall, at invocation, inform the user of the terminal deficiency that requires the <newline>  
 25842        character usage and provide instructions on how this warning can be suppressed in future  
 25843        invocations
- 25844        • Shall not be required for implementations supporting only fully capable terminals
- 25845        • Shall not affect commands already requiring <newline> characters
- 25846        • Shall not affect users on the capable terminals from using *more* as described in this volume of  
 25847        IEEE Std. 1003.1-200x

25848 **OPTIONS**

25849        The *more* utility shall conform to the System Interface Definitions volume of  
 25850        IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

25851        The following options shall be supported:

- 25852        **-c**        If a screen is to be written that has no lines in common with the current screen, or  
 25853        *more* is writing its first screen, *more* does not scroll the screen, but instead redraws  
 25854        each line of the screen in turn, from the top of the screen to the bottom. In addition,  
 25855        if *more* is writing its first screen, the screen is cleared. This option may be silently  
 25856        ignored on devices with insufficient terminal capabilities.
- 25857        **-e**        By default, *more* shall exit immediately after writing the last line of the last file in  
 25858        the argument list. If the **-e** option is specified:
  - 25859            1. If there is only a single file in the argument list and that file was completely  
 25860            displayed on a single screen, *more* shall exit immediately after writing the last  
 25861            line of that file.
  - 25862            2. Otherwise, *more* shall exit only after reaching end-of-file on the last file in the  
 25863            argument list twice without an intervening operation. See the EXTENDED  
 25864            DESCRIPTION section.
- 25865        **-i**        Perform pattern matching in searches without regard to case; see the System  
 25866        Interface Definitions volume of IEEE Std. 1003.1-200x, Section 9.2, Regular  
 25867        Expression General Requirements .

25868        **-n number**    Specify the number of lines per screenful. The *number* argument is a positive  
 25869                       decimal integer. The **-n** option shall override any values obtained from any other  
 25870                       source.

25871        **-p command**   Each time a screen from a new file is displayed or redisplayed (including as a  
 25872                       result of *more* commands; for example, **:p**), execute the *more* command(s) in the  
 25873                       command arguments in the order specified, as if entered by the user after the first  
 25874                       screen has been displayed. No intermediate results shall be displayed (that is, if the  
 25875                       command is a movement to a screen different than the normal first screen, only the  
 25876                       screen resulting from the command shall be displayed.) If any of the commands  
 25877                       fail for any reason, an informational message to this effect shall be written, and no  
 25878                       further commands specified using the **-p** option shall be executed for this file.

25879        **-s**                Behave as if consecutive empty lines were a single empty line.

25880        **-t tagstring**   Write the screenful of the file containing the tag named by the *tagstring* argument.  
 25881                       See the *ctags* utility. The tags feature represented by **-t tagstring** and the **:t**  
 25882                       command is optional. It shall be provided on any system that also provides a  
 25883                       conforming implementation of *ctags*; otherwise, the use of **-t** produces undefined  
 25884                       results.

25885                       The file name resulting from the **-t** option shall be logically added as a prefix to the  
 25886                       list of command line files, as if specified by the user. If the tag named by the  
 25887                       *tagstring* argument is not found, it shall be an error, and *more* shall take no further  
 25888                       action.

25889                       If the tag specifies a line number, the first line of the display shall contain the  
 25890                       beginning of that line. If the tag specifies a pattern, the first line of the display shall  
 25891                       contain the beginning of the matching text from the first line of the file that  
 25892                       contains that pattern. If the line does not exist in the file or matching text is not  
 25893                       found, an informational message to this effect shall be displayed, and *more* shall  
 25894                       display the default screen as if **-t** had not been specified.

25895                       If both the **-t tagstring** and **-p command** options are given, the **-t tagstring** shall be  
 25896                       processed first; that is, the file and starting line for the display shall be as specified  
 25897                       by **-t**, and then the **-p more** command shall be executed. If the line (matching text)  
 25898                       specified by the **-t** command does not exist (is not found), no **-p more** command  
 25899                       shall be executed for this file at any time.

25900        **-u**                Treat a <backspace> character as a printable control character, displayed as an  
 25901                       implementation-dependent character sequence (see the EXTENDED  
 25902                       DESCRIPTION section), suppressing backspacing and the special handling that  
 25903                       produces underlined or standout mode text on some terminal types. Also, do not  
 25904                       ignore a <carriage-return> character at the end of a line.

## 25905 OPERANDS

25906        The following operand shall be supported:

25907        *file*            A path name of an input file. If no *file* operands are specified, the standard input  
 25908                       shall be used. If a *file* is '-', the standard input shall be read at that point in the  
 25909                       sequence.

## 25910 STDIN

25911        The standard input shall be used only if no *file* operands are specified, or if a *file* operand is '-'.

## 25912 INPUT FILES

25913 **Notes to Reviewers**

25914 *This section with side shading will not appear in the final copy. - Ed.*

25915 D1, XCU, ERN 308 raises the issue of whether stderr is opened for input and output. There is  
 25916 also a corresponding ERN against XSH.

25917 The input files being examined shall be text files. If standard output is a terminal, standard error  
 25918 shall be used to read commands from the user. If standard output is a terminal, standard error is  
 25919 not readable, and command input is needed, *more* may attempt to obtain user commands from  
 25920 the controlling terminal (for example, **/dev/tty**); otherwise, *more* shall terminate with an error  
 25921 indicating that it was unable to read user commands. If standard output is not a terminal, no  
 25922 error shall result if standard error cannot be opened for reading.

25923 **ENVIRONMENT VARIABLES**

25924 The following environment variables shall affect the execution of *more*:

25925 **COLUMNS** Override the system-selected horizontal screen size. See the System Interface  
 25926 Definitions volume of IEEE Std. 1003.1-200x, Chapter 8, Environment Variables for  
 25927 valid values and results when it is unset or null.

25928 **EDITOR** Used by the **v** command to select an editor. See the EXTENDED DESCRIPTION  
 25929 section.

25930 **LANG** Provide a default value for the internationalization variables that are unset or null.  
 25931 If **LANG** is unset or null, the corresponding value from the implementation-  
 25932 dependent default locale shall be used. If any of the internationalization variables  
 25933 contains an invalid setting, the utility shall behave as if none of the variables had  
 25934 been defined.

25935 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 25936 internationalization variables.

25937 **LC\_COLLATE**  
 25938 Determine the locale for the behavior of ranges, equivalence classes, and multi-  
 25939 character collating elements within regular expressions.

25940 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 25941 characters (for example, single-byte as opposed to multi-byte characters in  
 25942 arguments and input files) and the behavior of character classes within regular  
 25943 expressions.

25944 **LC\_MESSAGES**  
 25945 Determine the locale that should be used to affect the format and contents of  
 25946 diagnostic messages written to standard error and informative messages written to  
 25947 standard output.

25948 XSI **NLSPATH** Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

25949 **LINES** Override the system-selected vertical screen size, used as the number of lines in a  
 25950 screenful. See the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 25951 Chapter 8, Environment Variables for valid values and results when it is unset or  
 25952 null. The **-n** option shall take precedence over the **LINES** variable for determining  
 25953 the number of lines in a screenful.

25954 **MORE** Determine a string containing options described in the OPTIONS section preceded  
 25955 with hyphens and <blank> character-separated as on the command line. Any  
 25956 command line options shall be processed after those in the **MORE** variable, as if

25957 the command line were:

25958 `more $MORE options operands`

25959 The *MORE* variable shall take precedence over the *TERM* and *LINES* variables for  
 25960 determining the number of lines in a screenful.

25961 *TERM* Determine the name of the terminal type. If this variable is unset or null, an  
 25962 unspecified default terminal type is used.

25963 **ASYNCHRONOUS EVENTS**

25964 Default.

25965 **STDOUT**

25966 The standard output shall be used to write the contents of the input files.

25967 **STDERR**

25968 Used for diagnostic messages and user commands (see the INPUT FILES section), and, if  
 25969 standard output is a terminal device, to write a prompting string. The prompting string shall  
 25970 shall appear on the screen line below the last line of the file displayed in the current screenful.  
 25971 The prompt shall contain the name of the file currently being examined and shall contain an  
 25972 end-of-file indication and the name of the next file, if any, when prompting at the end-of-file. If  
 25973 an error or informational message is displayed, it is unspecified whether it is contained in the  
 25974 prompt. If it is not contained in the prompt, it shall be displayed and then the user shall be  
 25975 prompted for a continuation character, at which point another message or the user prompt may  
 25976 be displayed. The prompt is otherwise unspecified. It is unspecified whether informational  
 25977 messages are written for other user commands.

25978 **OUTPUT FILES**

25979 None.

25980 **EXTENDED DESCRIPTION**

25981 The following subsection describes the behavior of *more* when the standard output is a terminal  
 25982 device. If the standard output is not a terminal device, no options other than `-s` shall have any  
 25983 effect, and all input files shall be copied to standard output otherwise unmodified, at which time  
 25984 *more* shall exit without further action.

25985 The number of lines available per screen shall be determined by the `-n` option, if present, or by  
 25986 examining values in the environment (see the ENVIRONMENT VARIABLES section). If neither  
 25987 method yields a number, an unspecified number of lines shall be used.

25988 The maximum number of lines written shall be one less than this number, because the screen  
 25989 line after the last line written shall be used to write a user prompt and user input. If the number  
 25990 of lines in the screen is less than two, the results are undefined. It is unspecified whether user  
 25991 input is permitted to be longer than the remainder of the single line where the prompt has been  
 25992 written.

25993 The number of columns available per line shall be determined by examining values in the  
 25994 environment (see the ENVIRONMENT VARIABLES section), with a default value as described  
 25995 in System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 8, Environment  
 25996 Variables.

25997 Lines that are longer than the display shall be folded; the length at which folding occurs is  
 25998 unspecified, but should be appropriate for the output device. Folding may occur between glyphs  
 25999 of single characters that take up multiple display columns.

26000 When standard output is a terminal and `-u` is not specified, *more* shall treat <backspace>  
 26001 characters and <carriage-return> characters specially:

- 26002 • A character, followed first by a sequence of *n* <backspace> characters (where *n* is the same as  
26003 the number of column positions that the character occupies), then by *n* underscore characters  
26004 ('\_'), shall cause that character to be written as underlined text, if the terminal type  
26005 supports that. The *n* underscore characters, followed first by *n* <backspace> characters, then  
26006 any character with *n* column positions, shall also cause that character to be written as  
26007 underlined text, if the terminal type supports that.
- 26008 • A sequence of *n* <backspace> characters (where *n* is the same as the number of column  
26009 positions that the previous character occupies) that appears between two identical printable  
26010 characters shall cause the first of those two characters to be written as emboldened text (that  
26011 is, visually brighter, standout mode, or inverse-video mode), if the terminal type supports  
26012 that, and the second to be discarded. Immediately subsequent occurrences of  
26013 <backspace>/character pairs for that same character also shall be discarded. (For example,  
26014 the sequence "a\ba\ba\ba" is interpreted as a single emboldened 'a'.)
- 26015 • The *more* utility shall logically discard all other <backspace> characters from the line as well  
26016 as the character which precedes them, if any.
- 26017 • A <carriage-return> character at the end of a line shall be ignored, rather than being written  
26018 as a non-printable character, as described in the next paragraph.

26019 It is implementation-dependent how other non-printable characters are written.  
26020 Implementations should use the same format that they use for the **ex print** command; see the  
26021 OPTIONS section within the *ed* utility. It is unspecified whether a multi-column character shall  
26022 be separated if it crosses a logical line boundary; it shall not be discarded. The behavior is  
26023 unspecified if the number of columns on the display is less than the number of columns any  
26024 single character in the line being displayed would occupy.

26025 When each new file is displayed (or redisplayed), *more* shall write the first screen of the file.  
26026 Once the initial screen has been written, *more* shall prompt for a user command. If the execution  
26027 of the user command results in a screen that has lines in common with the current screen, and  
26028 the device has sufficient terminal capabilities, *more* shall scroll the screen; otherwise, it is  
26029 unspecified whether the screen is scrolled or redrawn.

26030 For all files but the last (including standard input if no file was specified, and for the last file as  
26031 well, if the **-e** option was not specified), when *more* has written the last line in the file, *more* shall  
26032 prompt for a user command. This prompt shall contain the name of the next file as well as an  
26033 indication that *more* has reached end-of-file. If the user command is **f**, <control>-F, <space>, **j**,  
26034 <newline>, **d**, <control>-D, or **s**, *more* shall display the next file. Otherwise, if displaying the last  
26035 file, *more* shall exit. Otherwise, *more* shall execute the user command specified.

26036 Several of the commands described in this section display a previous screen from the input  
26037 stream. In the case that text is being taken from a non-rewindable stream, such as a pipe, it is  
26038 implementation-dependent how much backwards motion is supported. If a command cannot be  
26039 executed because of a limitation on backwards motion, an error message to this effect shall be  
26040 displayed, the current screen shall not change, and the user shall be prompted for another  
26041 command.

26042 If a command cannot be performed because there are insufficient lines to display, *more* shall alert  
26043 the terminal. If a command cannot be performed because there are insufficient lines to display or  
26044 a / command fails: if the input is the standard input, the last screen in the file may be displayed;  
26045 otherwise, the current file and screen shall not change, and the user shall be prompted for  
26046 another command.

26047 The interactive commands in the following sections shall be supported. Some commands can be  
26048 preceded by a decimal integer, called *count* in the following descriptions. If not specified with  
26049 the command, *count* shall default to 1. In the following descriptions, *pattern* is a basic regular

expression, as described in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 9.3, Basic Regular Expressions. The term “examine” is historical usage meaning “open the file for viewing”; for example, *more* **foo** would be expressed as examining file **foo**. In the following descriptions, unless otherwise specified, *line* is a logical line in the *more* display, not a line from the file being examined.

In the following descriptions, the *current position* refers to two things:

1. The position of the current line on the screen
2. The line number (in the file) of the current line on the screen

Usually, the line on the screen corresponding to the current position is the third line on the screen. If this is not possible (there are fewer than three lines to display or this is the first page of the file, or it is the last page of the file), then the current position is either the first or last line on the screen as described later.

## 26062 Help

26063 *Synopsis:*     **h**

26064 Write a summary of these commands and other implementation-dependent commands. The  
26065 behavior shall be as if the *more* utility were executed with the **-e** option on a file that contained  
26066 the summary information. The user shall be prompted as described earlier in this section when  
26067 end-of-file is reached. If the user command is one of those specified to continue to the next file,  
26068 *more* shall return to the file and screen state from which the **h** command was executed.

## 26069 Scroll Forward One Screenful

26070 *Synopsis:*     [ *count* ] **f**  
26071               [ *count* ] <control> -F

26072 Scroll forward *count* lines, with a default of one screenful. If *count* is more than the screen size,  
26073 only the final screenful shall be written.

## 26074 Scroll Backward One Screenful

26075 *Synopsis:*     [ *count* ] **b**  
26076               [ *count* ] <control> -B

26077 Scroll backward *count* lines, with a default of one screenful (see the **-n** option). If *count* is more  
26078 than the screen size, only the final screenful shall be written.

## 26079 Scroll Forward One Line

26080 *Synopsis:*     [ *count* ] <space>  
26081               [ *count* ] **j**  
26082               [ *count* ] <newline>

26083 Scroll forward *count* lines. The default *count* for the <space> character shall be one screenful; for **j**  
26084 and <newline> character, one line. The entire *count* lines shall be written, even if *count* is more  
26085 than the screen size.

**26086 Scroll Backward One Line**

26087 *Synopsis:*     [ *count* ]k

26088 Scroll backward *count* lines. The entire *count* lines shall be written, even if *count* is more than the  
26089 screen size.

**26090 Scroll Forward One Half Screenful**

26091 *Synopsis:*     [ *count* ]d

26092               [ *count* ]<control>-D

26093 Scroll forward *count* lines, with a default of one half of the screen size. If *count* is specified, it  
26094 shall become the new default for subsequent **d**, <control>-D, and **u** commands.

**26095 Skip Forward One Line**

26096 *Synopsis:*     [ *count* ]s

26097 Display the screenful beginning with the line *count* lines after the last line on the current screen.  
26098 If *count* would cause the current position to be such that less than one screenful would be  
26099 written, the last screenful in the file shall be written.

**26100 Scroll Backward One Half Screenful**

26101 *Synopsis:*     [ *count* ]u

26102               [ *count* ]<control>-U

26103 Scroll backward *count* lines, with a default of one half of the screen size. If *count* is specified, it  
26104 shall become the new default for subsequent **d**, <control>-D, **u**, and <control>-U commands.  
26105 The entire *count* lines shall be written, even if *count* is more than the screen size.

**26106 Go to Beginning of File**

26107 *Synopsis:*     [ *count* ]g

26108 Display the screenful beginning with line *count*.

**26109 Go to End-of-File**

26110 *Synopsis:*     [ *count* ]G

26111 If *count* is specified, display the screenful beginning with the line *count*. Otherwise, display the  
26112 last screenful of the file.

**26113 Refresh the Screen**

26114 *Synopsis:*     r

26115               <control>-L

26116 Refresh the screen.

## 26117 **Discard and Refresh**

26118 *Synopsis:*     **R**

26119 Refresh the screen, discarding any buffered input. If the current file is non-seekable, buffered  
26120 input shall not be discarded and the **R** command is equivalent to the **r** command.

## 26121 **Mark Position**

26122 *Synopsis:*     **mletter**

26123 Mark the current position with the letter named by *letter*, where *letter* represents the name of one  
26124 of the lowercase letters of the portable character set. When a new file is examined, all marks may  
26125 be lost.

## 26126 **Return to Mark**

26127 *Synopsis:*     '**letter**

26128 Return to the position that was previously marked with the letter named by *letter*, making that  
26129 line the current position.

## 26130 **Return to Previous Position**

26131 *Synopsis:*     ''

26132 Return to the position from which the last large movement command was executed (where a  
26133 “large movement” is defined as any movement of more than a screenful of lines). If no such  
26134 movements have been made, return to the beginning of the file.

## 26135 **Search Forward for Pattern**

26136 *Synopsis:*     [*count*]/[!]*pattern*<newline>

26137 Display the screenful beginning with the *count*th line containing the pattern. The search shall  
26138 start after the first line currently displayed. The null regular expression ('/' followed by a  
26139 <newline> character) shall repeat the search using the previous regular expression, with a  
26140 default *count*. If the character '!' is included, the matching lines shall be those that do not  
26141 contain the *pattern*. If no match is found for the *pattern*, a message to that effect shall be  
26142 displayed.

## 26143 **Search Backward for Pattern**

26144 *Synopsis:*     [*count*?][!]*pattern*<newline>

26145 Display the screenful beginning with the *count*th previous line containing the pattern. The  
26146 search shall start on the last line before the first line currently displayed. The null regular  
26147 expression ('?' followed by a <newline> character) shall repeat the search using the previous  
26148 regular expression, with a default *count*. If the character '!' is included, matching lines shall be  
26149 those that do not contain the *pattern*.

26150 If no match is found for the *pattern*, a message to that effect shall be displayed.



**26151 Repeat Search**

26152 *Synopsis:*     [ *count* ]n

26153 Repeat the previous search for *count*th line containing the last *pattern* (or not containing the last  
26154 *pattern*, if the previous search was " / ! " or " ? ! " ).

**26155 Repeat Search in Reverse**

26156 *Synopsis:*     [ *count* ]N

26157 Repeat the search in the opposite direction of the previous search for the *count*th line containing  
26158 the last *pattern* (or not containing the last *pattern*, if the previous search was " / ! " or " ? ! " ).

**26159 Examine New File**

26160 *Synopsis:*     :e [ *filename* ]<newline>

26161 Examine a new file. If the *filename* argument is not specified, the current file (see the :n and :p  
26162 commands below) shall be re-examined. The *filename* shall be subjected to the process of shell  
26163 word expansions (see Section 2.6 on page 49); if more than a single path name results, the effects  
26164 are unspecified. If *filename* is a number sign ( '#' ), the previously examined file shall be re-  
26165 examined. If *filename* is not accessible for any reason (including that it is a non-seekable file), an  
26166 error message to this effect shall be displayed and the current file and screen shall not change.

**26167 Examine Next File**

26168 *Synopsis:*     [ *count* ]:n

26169 Examine the next file. If a number *count* is specified, the *count*th next file shall be examined. If  
26170 *filename* refers to a non-seekable file, the results are unspecified.

**26171 Examine Previous File**

26172 *Synopsis:*     [ *count* ]:p

26173 Examine the previous file. If a number *count* is specified, the *count*th previous file shall be  
26174 examined. If *filename* refers to a non-seekable file, the results are unspecified.

**26175 Go to Tag**

26176 *Synopsis:*     :t *tagstring*<newline>

26177 If the file containing the tag named by the *tagstring* argument is not the current file, examine the  
26178 file, as if the :e command was executed with that file as the argument. Otherwise, or in addition,  
26179 display the screenful beginning with the tag, as described for the -t option (see the OPTIONS  
26180 section). If the *ctags* utility is not supported by the system, the use of :t produces undefined  
26181 results.

**26182 Invoke Editor**

26183 *Synopsis:*     v

26184 Invoke an editor to edit the current file being examined. If standard input is being examined, the  
26185 results are unspecified. The name of the editor shall be taken from the environment variable  
26186 *EDITOR*, or shall default to *vi*. If the last path name component in *EDITOR* is either *vi* or *ex*, the  
26187 editor shall be invoked with a - *linenumber* command line argument, where *linenumber* is the line  
26188 number of the physical line containing the logical line currently displayed as the first line of the  
26189 screen. It is implementation-dependent whether line-setting options are passed to editors other

26190 than *vi* and *ex*.  
 26191 When the editor exits, *more* shall resume with the same file and screen as when the editor was  
 26192 invoked.

### 26193 **Display Position**

26194 *Synopsis:*        =  
 26195                    <control>-G

26196 Write a message for which the information references the first byte of the line after the last line of  
 26197 the file on the screen. This message shall include the name of the file currently being examined,  
 26198 its number relative to the total number of files there are to examine, the physical line number,  
 26199 the byte number and the total bytes in the file, and what percentage of the file precedes the  
 26200 current position. If *more* is reading from standard input, or the file is shorter than a single screen,  
 26201 the line number, the byte number, the total bytes, and the percentage need not be written.

### 26202 **Quit**

26203 *Synopsis:*        q  
 26204                   :q  
 26205                   ZZ

26206 Exit *more*.

### 26207 **EXIT STATUS**

26208 The following exit values shall be returned:

26209       0   Successful completion.

26210       >0   An error occurred.

### 26211 **CONSEQUENCES OF ERRORS**

26212 If an error is encountered accessing a file when using the **:n** command, *more* shall attempt to  
 26213 examine the next file in the argument list, but the final exit status shall be affected. If an error is  
 26214 encountered accessing a file via the **:p** command, *more* shall attempt to examine the previous file  
 26215 in the argument list, but the final exit status shall be affected. If an error is encountered accessing  
 26216 a file via the **:e** command, *more* shall remain in the current file and the final exit status shall not  
 26217 be affected.

### 26218 **APPLICATION USAGE**

26219 When the standard output is not a terminal, only the **-s** filter-modification option is effective.  
 26220 This is based on historical practice. For example, a typical implementation of *man* pipes its  
 26221 output through *more -s* to squeeze excess white space for terminal users. When *man* is piped to  
 26222 *lp*, however, it is undesirable for this squeezing to happen.

26223 Application writers should note that this utility need not be provided on systems that do not  
 26224 support the User Portability Utilities option.

### 26225 **EXAMPLES**

26226 The **-p** allows arbitrary commands to be executed at the start of each file. Examples are:

26227 *more -p G file1 file2*  
 26228       Examine each file starting with its last screenful.

26229 *more -p 100 file1 file2*  
 26230       Examine each file starting with line 100 in the current position (usually the third line, so line  
 26231 98 would be the first line written).

26232 *more* -p /100 file1 file2

26233 Examine each file starting with the first line containing the string "100" in the current  
26234 position

#### 26235 RATIONALE

26236 The *more* utility, available in BSD and BSD-derived systems, was chosen as the prototype for the  
26237 POSIX file display program since it is more widely available than either the public-domain  
26238 program *less* or than *pg*, a pager provided in System V. The 4.4 BSD *more* is the model for the  
26239 features selected; it is almost fully upward-compatible from the 4.3 BSD version in wide use and  
26240 has become more amenable for *vi* users. Several features originally derived from various file  
26241 editors, found in both *less* and *pg*, have been added to this volume of IEEE Std. 1003.1-200x as  
26242 they have proved extremely popular with users.

26243 There are inconsistencies between *more* and *vi* that result from historical practice. For example,  
26244 the single-character commands **h**, **f**, **b**, and <space> are screen movers in *more*, but cursor  
26245 movers in *vi*. These inconsistencies were maintained because the cursor movements are not  
26246 applicable to *more* and the powerful functionality achieved without the use of the control key  
26247 justifies the differences.

26248 The tags interface has been included in a program that is not a text editor because it promotes  
26249 another degree of consistent operation with *vi*. It is conceivable that the paging environment of  
26250 *more* would be superior for browsing source code files in some circumstances.

26251 The operating mode referred to for block-mode terminals effectively adds a <newline> to each  
26252 Synopsis line that currently has none. So, for example, **d**<newline> would page one screenful.  
26253 The mode could be triggered by a command line option, environment variable, or some other  
26254 method. The details are not imposed by this volume of IEEE Std. 1003.1-200x because there are  
26255 so few systems known to support such terminals. Nevertheless, it was considered that all  
26256 systems should be able to support *more* given the exception cited for this small community of  
26257 terminals because, in comparison to *vi*, the cursor movements are few and the command set  
26258 relatively amenable to the optional <newline>s.

26259 Some versions of *more* provide a shell escaping mechanism similar to the *ex* ! command. The  
26260 standard developers did not consider that this was necessary in a paginator, particularly given  
26261 the wide acceptance of multiple window terminals and job control features. (They chose to  
26262 retain such features in the editors and *mailx* because the shell interaction also gives an  
26263 opportunity to modify the editing buffer, which is not applicable to *more*).

26264 The -p (position) option replaces the + command because of the Utility Syntax Guidelines. In  
26265 early proposals, it took a *pattern* argument, but historical *less* provided the *more* general facility of  
26266 a command. It would have been desirable to use the same -c as *ex* and *vi*, but the letter was  
26267 already in use.

26268 The text stating "from a non-rewindable stream ... implementations may limit the amount of  
26269 backwards motion supported" would allow an implementation that permitted no backwards  
26270 motion beyond text already on the screen. It was not possible to require a minimum amount of  
26271 backwards motion that would be effective for all conceivable device types. The implementation  
26272 should allow the user to back up as far as possible, within device and reasonable memory  
26273 allocation constraints.

26274 Historically, non-printable characters were displayed using the ARPA standard mappings,  
26275 which are as follows:

- 26276 1. Printable characters are left alone.
- 26277 2. Control characters less than \177 are represented as followed by the character offset from  
26278 the '@' character in the ASCII map; for example, \007 is represented as 'G'.

|       |                                                                                                                                                       |  |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------|--|
| 26279 | 3. \177 is represented as followed by ' ? '.                                                                                                          |  |
| 26280 | The display of characters having their eighth bit set was less standard. Existing implementations                                                     |  |
| 26281 | use hex (0x00), octal (\000), and a meta-bit display. (The latter displayed characters with their                                                     |  |
| 26282 | eighth bit set as the two characters "M-, " followed by the seven bit display as described                                                            |  |
| 26283 | previously.) The latter probably has the best claim to historical practice because it was used with                                                   |  |
| 26284 | the -v option of 4 BSD and 4 BSD-derived versions of the <i>cat</i> utility since 1980.                                                               |  |
| 26285 | No specific display format is required by IEEE Std. 1003.1-200x. Implementations are                                                                  |  |
| 26286 | encouraged to conform to historic practice in the absence of any strong reason to diverge.                                                            |  |
| 26287 | <b>FUTURE DIRECTIONS</b>                                                                                                                              |  |
| 26288 | None.                                                                                                                                                 |  |
| 26289 | <b>SEE ALSO</b>                                                                                                                                       |  |
| 26290 | <i>ctags, ed, ex, vi</i>                                                                                                                              |  |
| 26291 | <b>CHANGE HISTORY</b>                                                                                                                                 |  |
| 26292 | First released in Issue 4.                                                                                                                            |  |
| 26293 | <b>Issue 5</b>                                                                                                                                        |  |
| 26294 | FUTURE DIRECTIONS section added.                                                                                                                      |  |
| 26295 | <b>Issue 6</b>                                                                                                                                        |  |
| 26296 | This utility is now marked as part of the User Portability Utilities option.                                                                          |  |
| 26297 | The obsolescent SYNOPSIS is removed.                                                                                                                  |  |
| 26298 | The utility has been extensively reworked for alignment with the IEEE P1003.2b draft standard:                                                        |  |
| 26299 | <ul style="list-style-type: none"> <li>• Changes have been made as result of PASC Interpretations 1003.2-92 #37 and 109.</li> </ul>                   |  |
| 26300 | <ul style="list-style-type: none"> <li>• The <i>more</i> utility should be able to handle underlined and emboldened displays of characters</li> </ul> |  |
| 26301 | that are wider than a single column position.                                                                                                         |  |

26302 **NAME**

26303 mv — move files

26304 **SYNOPSIS**26305 mv [-fi] *source\_file target\_file*26306 mv [-fi] *source\_file... target\_file*26307 **DESCRIPTION**

26308 In the first synopsis form, the *mv* utility shall move the file named by the *source\_file* operand to  
 26309 the *destination* specified by the *target\_file*. This first synopsis form is assumed when the final  
 26310 operand does not name an existing directory and is not a symbolic link referring to an existing  
 26311 directory.

26312 In the second synopsis form, *mv* shall move each file named by a *source\_file* operand to a  
 26313 *destination* file in the existing directory named by the *target\_dir* operand, or referenced if  
 26314 *target\_dir* is a symbolic link referring to an existing directory. The *destination* path for each  
 26315 *source\_file* shall be the concatenation of the target directory, a single slash character, and the last  
 26316 path name component of the *source\_file*. This second form is assumed when the final operand  
 26317 names an existing directory.

26318 If any operand specifies an existing file of a type not specified by the System Interfaces volume  
 26319 of IEEE Std. 1003.1-200x, the behavior is implementation-dependent.

26320 For each *source\_file* the following steps shall be taken:

26321 1. If the destination path exists, the *-f* option is not specified, and either of the following  
 26322 conditions is true:

26323 a. The permissions of the destination path do not permit writing and the standard input  
 26324 is a terminal.

26325 b. The *-i* option is specified.

26326 The *mv* utility shall write a prompt to standard error and read a line from standard input. If  
 26327 the response is not affirmative, *mv* shall do nothing more with the current *source\_file* and  
 26328 go on to any remaining *source\_files*.

26329 2. The *mv* utility shall perform actions equivalent to the *rename()* function defined in the  
 26330 System Interfaces volume of IEEE Std. 1003.1-200x, called with the following arguments:

26331 a. The *source\_file* operand is used as the *old* argument.

26332 b. The destination path is used as the *new* argument.

26333 If this succeeds, *mv* shall do nothing more with the current *source\_file* and go on to any  
 26334 remaining *source\_files*. If this fails for any reasons other than those described for the *errno*  
 26335 [EXDEV] in the System Interfaces volume of IEEE Std. 1003.1-200x, *mv* shall write a  
 26336 diagnostic message to standard error, do nothing more with the current *source\_file*, and go  
 26337 on to any remaining *source\_files*.

26338 3. If the destination path exists, and it is a file of type directory and *source\_file* is not a file of  
 26339 type directory, or it is a file not of type directory and *source\_file* is a file of type directory,  
 26340 *mv* shall write a diagnostic message to standard error, do nothing more with the current  
 26341 *source\_file*, and go on to any remaining *source\_files*.

26342 4. If the destination path exists, *mv* shall attempt to remove it. If this fails for any reason, *mv*  
 26343 shall write a diagnostic message to standard error, do nothing more with the current  
 26344 *source\_file*, and go on to any remaining *source\_files*.

- 26345 5. The file hierarchy rooted in *source\_file* shall be duplicated as a file hierarchy rooted in the  
 26346 *destination* path. If *source\_file* or any of the files below it in the hierarchy are symbolic links,  
 26347 the links themselves shall be duplicated, including their contents, rather than any files to  
 26348 which they refer. The following characteristics of each file in the file hierarchy shall be  
 26349 duplicated:
- 26350 • The time of last data modification and time of last access
  - 26351 • The user ID and group ID
  - 26352 • The file mode
- 26353 If the user ID, group ID, or file mode of a regular file cannot be duplicated, the file mode  
 26354 bits S\_ISUID and S\_ISGID shall not be duplicated.
- 26355 When files are duplicated to another file system, the implementation may require that the  
 26356 process invoking *mv* has read access to each file being duplicated.
- 26357 If the duplication of the file hierarchy fails for any reason, *mv* shall write a diagnostic  
 26358 message to standard error, do nothing more with the current *source\_file*, and go on to any  
 26359 remaining *source\_files*.
- 26360 If the duplication of the file characteristics fails for any reason, *mv* shall write a diagnostic  
 26361 message to standard error, but this failure shall not cause *mv* to modify its exit status.
- 26362 6. The file hierarchy rooted in *source\_file* shall be removed. If this fails for any reason, *mv* shall  
 26363 write a diagnostic message to the standard error, do nothing more with the current  
 26364 *source\_file*, and go on to any remaining *source\_files*.

#### 26365 OPTIONS

- 26366 The *mv* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 26367 Section 12.2, Utility Syntax Guidelines.
- 26368 The following options shall be supported:
- 26369 **-f** Do not prompt for confirmation if the destination path exists. Any previous  
 26370 occurrences of the **-i** option is ignored.
  - 26371 **-i** Prompt for confirmation if the destination path exists. Any previous occurrences of  
 26372 the **-f** option is ignored.
- 26373 Specifying more than one of the **-f** or **-i** options shall not be considered an error. The last option  
 26374 specified shall determine the behavior of *mv*.

#### 26375 OPERANDS

- 26376 The following operands shall be supported:
- 26377 *source\_file* A path name of a file or directory to be moved.
  - 26378 *target\_file* A new path name for the file or directory being moved.
  - 26379 *target\_dir* A path name of an existing directory into which to move the input files.

#### 26380 STDIN

- 26381 Used to read an input line in response to each prompt specified in the STDERR section.  
 26382 Otherwise, the standard input shall not be used.

#### 26383 INPUT FILES

- 26384 The input files specified by each *source\_file* operand can be of any file type.

26385 **ENVIRONMENT VARIABLES**

26386 The following environment variables shall affect the execution of *mv*:

26387 **LANG** Provide a default value for the internationalization variables that are unset or null.  
 26388 If *LANG* is unset or null, the corresponding value from the implementation-  
 26389 dependent default locale shall be used. If any of the internationalization variables  
 26390 contains an invalid setting, the utility shall behave as if none of the variables had  
 26391 been defined.

26392 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 26393 internationalization variables.

26394 **LC\_COLLATE**  
 26395 Determine the locale for the behavior of ranges, equivalence classes and multi-  
 26396 character collating elements used in the extended regular expression defined for  
 26397 the **yesexpr** locale keyword in the *LC\_MESSAGES* category.

26398 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 26399 characters (for example, single-byte as opposed to multi-byte characters in  
 26400 arguments and input files), the behavior of character classes used in the extended  
 26401 regular expression defined for the **yesexpr** locale keyword in the *LC\_MESSAGES*  
 26402 category.

26403 **LC\_MESSAGES**  
 26404 Determine the locale for the processing of affirmative responses that should be  
 26405 used to affect the format and contents of diagnostic messages written to standard  
 26406 error.

26407 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

26408 **ASYNCHRONOUS EVENTS**

26409 Default.

26410 **STDOUT**

26411 Not used.

26412 **STDERR**

26413 Prompts shall be written to the standard error under the conditions specified in the  
 26414 DESCRIPTION section. The prompts shall contain the *destination* path name, but their format is  
 26415 otherwise unspecified. Otherwise, the standard error shall be used only for diagnostic messages.

26416 **OUTPUT FILES**

26417 The output files may be of any file type.

26418 **EXTENDED DESCRIPTION**

26419 None.

26420 **EXIT STATUS**

26421 The following exit values shall be returned:

26422 0 All input files were moved successfully.

26423 >0 An error occurred.

26424 **CONSEQUENCES OF ERRORS**

26425 If the copying or removal of *source\_file* is prematurely terminated by a signal or error, *mv* may  
 26426 leave a partial copy of *source\_file* at the source or destination. The *mv* utility shall not modify  
 26427 both *source\_file* and the destination path simultaneously; termination at any point shall leave  
 26428 either *source\_file* or the destination path complete.

## 26429 APPLICATION USAGE

26430 None.

## 26431 EXAMPLES

26432 If the current directory contains only files **a** (of any type defined by the System Interfaces  
 26433 volume of IEEE Std. 1003.1-200x), **b** (also of any type), and a directory **c**:

26434 mv a b c

26435 mv c d

26436 results with the original files **a** and **b** residing in the directory **d** in the current directory.

## 26437 RATIONALE

26438 Early proposals diverged from the SVID and BSD historical practice in that they required that  
 26439 when the destination path exists, the **-f** option is not specified, and input is not a terminal, *mv*  
 26440 fails. This was done for compatibility with *cp*. The current text returns to historical practice. It  
 26441 should be noted that this is consistent with the *rename()* function defined in the System  
 26442 Interfaces volume of IEEE Std. 1003.1-200x, which does not require write permission on the  
 26443 target.

26444 For absolute clarity, paragraph (1), describing the behavior of *mv* when prompting for  
 26445 confirmation, should be interpreted in the following manner:

```
26446 if (exists AND (NOT f_option) AND
26447 ((not_writable AND input_is_terminal) OR i_option))
```

26448 The **-i** option exists on BSD systems, giving applications and users a way to avoid accidentally  
 26449 unlinking files when moving others. When the standard input is not a terminal, the 4.3 BSD *mv*  
 26450 deletes all existing destination paths without prompting, even when **-i** is specified; this is  
 26451 inconsistent with the behavior of the 4.3 BSD *cp* utility, which always generates an error when  
 26452 the file is unwritable and the standard input is not a terminal. The standard developers decided  
 26453 that use of **-i** is a request for interaction, so when the *destination* path exists, the utility takes  
 26454 instructions from whatever responds to standard input.

26455 The *rename()* function is able to move directories within the same file system. Some historical  
 26456 versions of *mv* have been able to move directories, but not to a different file system. The  
 26457 standard developers considered that this was an annoying inconsistency, so this volume of  
 26458 IEEE Std. 1003.1-200x requires directories to be able to be moved even across file systems. There  
 26459 is no **-R** option to confirm that moving a directory is actually intended, since such an option was  
 26460 not required for moving directories in historical practice. Requiring the application to specify it  
 26461 sometimes, depending on the destination, seemed just as inconsistent. The semantics of the  
 26462 *rename()* function were preserved as much as possible. For example, *mv* is not permitted to  
 26463 “rename” files to or from directories, even though they might be empty and removable.

26464 Historic implementations of *mv* did not exit with a non-zero exit status if they were unable to  
 26465 duplicate any file characteristics when moving a file across file systems, nor did they write a  
 26466 diagnostic message for the user. The former behavior has been preserved to prevent scripts from  
 26467 breaking; a diagnostic message is now required, however, so that users are alerted that the file  
 26468 characteristics have changed.

26469 The exact format of the interactive prompts is unspecified. Only the general nature of the  
 26470 contents of prompts are specified because implementations may desire more descriptive  
 26471 prompts than those used on historical implementations. Therefore, an application not using the  
 26472 **-f** option or using the **-i** option relies on the system to provide the most suitable dialog directly  
 26473 with the user, based on the behavior specified.

26474 When *mv* is dealing with a single file system and *source\_file* is a symbolic link, the link itself is  
 26475 moved as a consequence of the dependence on the *rename()* functionality, per the



26476 DESCRIPTION. Across file systems, this has to be made explicit.

26477 **FUTURE DIRECTIONS**

26478 None.

26479 **SEE ALSO**

26480 *cp, ln*

26481 **CHANGE HISTORY**

26482 First released in Issue 2.

26483 **Issue 4**

26484 Aligned with the ISO/IEC 9945-2:1993 standard.

26485 **Issue 6**

26486 The *mv* utility is changed to describe processing of symbolic links as specified in the  
26487 IEEE P1003.2b draft standard.

## 26488 NAME

26489 newgrp — change to a new group

## 26490 SYNOPSIS

26491 UP newgrp [-l][group]

26492

## 26493 DESCRIPTION

26494 The *newgrp* utility shall create a new shell execution environment with a new real and effective  
 26495 group identification. Of the attributes listed in Section 2.12 on page 90, the new shell execution  
 26496 environment shall retain the working directory, file creation mask, and exported variables from  
 26497 the previous environment (that is, open files, traps, unexported variables, alias definitions, shell  
 26498 functions, and *set* options may be lost). All other aspects of the process environment that are  
 26499 preserved by the *exec* family of functions defined in the System Interfaces volume of  
 26500 IEEE Std. 1003.1-200x shall also be preserved by *newgrp*; whether other aspects are preserved is  
 26501 unspecified.

26502 A failure to assign the new group identifications (for example, for security or password-related  
 26503 reasons) shall not prevent the new shell execution environment from being created.

26504 The *newgrp* utility shall affect the supplemental groups for the process as follows:

- 26505 • On systems where the effective group ID is normally in the supplementary group list (or  
 26506 whenever the old effective group ID actually is in the supplementary group list):
  - 26507 — If the new effective group ID is also in the supplementary group list, *newgrp* shall change  
 26508 the effective group ID.
  - 26509 — If the new effective group ID is not in the supplementary group list, *newgrp* shall add the  
 26510 new effective group ID to the list, if there is room to add it.
- 26511 • On systems where the effective group ID is not normally in the supplementary group list (or  
 26512 whenever the old effective group ID is not in the supplementary group list):
  - 26513 — If the new effective group ID is in the supplementary group list, *newgrp* shall delete it.
  - 26514 — If the old effective group ID is not in the supplementary list, *newgrp* shall add it if there is  
 26515 room.

26516 **Note:** The System Interfaces volume of IEEE Std. 1003.1-200x does not specify whether the  
 26517 effective group ID of a process is included in its supplementary group list.

26518 With no operands, *newgrp* shall change the effective group back to the groups identified in the  
 26519 user's user entry, and shall set the list of supplementary groups to that set in the user's group  
 26520 database entries.

26521 If a password is required for the specified group, and the user is not listed as a member of that  
 26522 group in the group database, the user shall be prompted to enter the correct password for that  
 26523 group. If the user is listed as a member of that group, no password is requested. If no password  
 26524 is required for the specified group, it is implementation-dependent whether users not listed as  
 26525 members of that group can change to that group. Whether or not a password is required,  
 26526 implementation-dependent system accounting or security mechanisms may impose additional  
 26527 authorization restrictions that may cause *newgrp* to write a diagnostic message and suppress the  
 26528 changing of the group identification.

## 26529 OPTIONS

26530 The *newgrp* utility shall conform to the System Interface Definitions volume of  
 26531 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

26532 The following option shall be supported:

26533 **-l** (The letter ell.) Change the environment to what would be expected if the user  
26534 actually logged in again.

#### 26535 OPERANDS

26536 The following operand shall be supported:

26537 *group* A group name from the group database or a non-negative numeric group ID.  
26538 Specifies the group ID to which the real and effective group IDs shall be set. If  
26539 *group* is a non-negative numeric string and exists in the group database as a group  
26540 name (see *getgrnam()*), the numeric group ID associated with that group name  
26541 shall be used as the group ID.

#### 26542 STDIN

26543 Not used.

#### 26544 INPUT FILES

26545 The file */dev/tty* shall be used to read a single line of text for password checking, when one is  
26546 required.

#### 26547 ENVIRONMENT VARIABLES

26548 The following environment variables shall affect the execution of *newgrp*:

26549 *LANG* Provide a default value for the internationalization variables that are unset or null.  
26550 If *LANG* is unset or null, the corresponding value from the implementation-  
26551 dependent default locale shall be used. If any of the internationalization variables  
26552 contains an invalid setting, the utility shall behave as if none of the variables had  
26553 been defined.

26554 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
26555 internationalization variables.

26556 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
26557 characters (for example, single-byte as opposed to multi-byte characters in  
26558 arguments).

#### 26559 LC\_MESSAGES

26560 Determine the locale that should be used to affect the format and contents of  
26561 diagnostic messages written to standard error.

26562 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

#### 26563 ASYNCHRONOUS EVENTS

26564 Default.

#### 26565 STDOUT

26566 Not used.

#### 26567 STDERR

26568 Used for diagnostic messages and a prompt string for a password, if one is required. Diagnostic  
26569 messages may be written in cases where the exit status is not available. See the EXIT STATUS  
26570 section.

#### 26571 OUTPUT FILES

26572 None.

26573 **EXTENDED DESCRIPTION**

26574       None.

26575 **EXIT STATUS**

26576       If *newgrp* succeeds in creating a new shell execution environment, whether or not the group  
26577       identification was changed successfully, the exit status shall be the exit status of the shell.  
26578       Otherwise, the following exit value shall be returned:

26579       &gt;0   An error occurred.

26580 **CONSEQUENCES OF ERRORS**

26581       The invoking shell may terminate.

26582 **APPLICATION USAGE**

26583       There is no convenient way to enter a password into the Group Database. Use of group  
26584       passwords is not encouraged, because by their very nature they encourage poor security  
26585       practices. Group passwords may disappear in the future.

26586       A common implementation of *newgrp* is that the current shell uses *exec* to overlay itself with  
26587       *newgrp*, which in turn overlays itself with a new shell after changing group. On some systems,  
26588       however, this may not occur and *newgrp* may be invoked as a subprocess.

26589       The *newgrp* command is intended only for use from an interactive terminal. It does not offer a  
26590       useful interface for the support of applications.

26591       The exit status of *newgrp* is generally inapplicable. If *newgrp* is used in a script, in most cases it  
26592       successfully invokes a new shell and the rest of the original shell script is bypassed when the  
26593       new shell exits. Used interactively, *newgrp* displays diagnostic messages to indicate problems.  
26594       But usage such as:

26595       newgrp foo  
26596       echo \$?

26597       is not useful because the new shell might not have access to any status *newgrp* may have  
26598       generated (and most historical systems do not provide this status). A zero status echoed here  
26599       does not necessarily indicate that the user has changed to the new group successfully. Following  
26600       *newgrp* with the *id* command provides a portable means of determining whether the group  
26601       change was successful or not.

26602       Application writers should note that this utility need not be provided on systems that do not  
26603       support the User Portability Utilities option.

26604 **EXAMPLES**

26605       None.

26606 **RATIONALE**

26607       Most historical implementations use one of the *exec* functions to implement the behavior of  
26608       *newgrp*. Errors detected before the *exec* leave the environment unchanged, while errors detected  
26609       after the *exec* leave the user in a changed environment. While it would be useful to have *newgrp*  
26610       issue a diagnostic message to tell the user that the environment changed, it would be  
26611       inappropriate to require this change to some historical implementations.

26612       The password mechanism is allowed in the group database, but how this would be  
26613       implemented is not specified.

26614       The *newgrp* utility was retained in this volume of IEEE Std. 1003.1-200x, even given the existence  
26615       of the multiple group permissions feature in the System Interfaces volume of  
26616       IEEE Std. 1003.1-200x, for several reasons. First, in some systems, the group ownership of a  
26617       newly created file is determined by the group of the directory in which the file is created, as

26618 allowed by the System Interfaces volume of IEEE Std. 1003.1-200x; on other systems, the group  
26619 ownership of a newly created file is determined by the effective group ID. On systems of the  
26620 latter type, *newgrp* allows files to be created with a specific group ownership. Finally, many  
26621 systems use the real group ID in accounting, and on such systems, *newgrp* allows the accounting  
26622 identity of the user to be changed.

26623 **FUTURE DIRECTIONS**

26624 None.

26625 **SEE ALSO**

26626 *sh*, the System Interfaces volume of IEEE Std. 1003.1-200x, *exec*

26627 **CHANGE HISTORY**

26628 First released in Issue 2.

26629 **Issue 4**

26630 Aligned with the ISO/IEC 9945-2: 1993 standard.

26631 The *newgrp* utility is now mandatory; it is optional in Issue 3.

26632 **Issue 6**

26633 This utility is now marked as part of the User Portability Utilities option.

26634 The obsolescent SYNOPSIS is removed.

26635 The text describing supplemental groups is no longer conditional on {NGROUPS\_MAX} being  
26636 greater than 1. This is because {NGROUPS\_MAX} now has a minimum value of 8. This is a FIPS  
26637 requirement.

26638 **NAME**

26639       *nice* — invoke a utility with an altered nice value

26640 **SYNOPSIS**

26641 UP       *nice* [*-n increment*] *utility* [*argument...*]

26642

26643 **DESCRIPTION**

26644       The *nice* utility shall invoke a utility, requesting that it be run with a different nice value (see the  
26645       System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 3.245, Nice Value). With  
26646       no options and only if the user has appropriate privileges, the executed utility shall be run with a  
26647       nice value that is some implementation-dependent quantity less than or equal to the nice value  
26648       of the current process. If the user lacks appropriate privileges to affect the nice value in the  
26649       requested manner, the *nice* utility shall not affect the nice value; in this case, a warning message  
26650       may be written to standard error, but this shall not prevent the invocation of *utility* or affect the  
26651       exit status.

26652 **OPTIONS**

26653       The *nice* utility shall conform to the System Interface Definitions volume of  
26654       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

26655       The following option is supported:

26656       *-n increment* Specify how the nice value of the executed utility shall be adjusted. The *increment*  
26657       option-argument shall be a positive or negative decimal integer that shall be used  
26658       to modify the nice value of the executed utility in an implementation-dependent  
26659       manner.

26660       Positive *increment* values shall cause a lower or unchanged nice value. Negative  
26661       *increment* values may require appropriate privileges and shall cause a higher or  
26662       unchanged nice value.

26663       The nice value shall be bounded in an implementation-dependent manner. If the  
26664       requested *increment* would raise or lower the nice value of the executed utility  
26665       beyond implementation-dependent limits, then the limit whose value was  
26666       exceeded shall be used.

26667 **OPERANDS**

26668       The following operands shall be supported:

26669       *utility*       The name of a utility that is to be invoked. If the *utility* operand names any of the  
26670       special built-in utilities in Section 2.14 on page 96, the results are undefined.

26671       *argument*     Any string to be supplied as an argument when invoking the utility named by the  
26672       *utility* operand.

26673 **STDIN**

26674       Not used.

26675 **INPUT FILES**

26676       None.

26677 **ENVIRONMENT VARIABLES**

26678       The following environment variables shall affect the execution of *nice*:

26679       *LANG*       Provide a default value for the internationalization variables that are unset or null.  
26680       If *LANG* is unset or null, the corresponding value from the implementation-  
26681       dependent default locale shall be used. If any of the internationalization variables  
26682       contains an invalid setting, the utility shall behave as if none of the variables had

|       |                                                 |                                                                                                                                  |
|-------|-------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------|
| 26683 |                                                 | been defined.                                                                                                                    |
| 26684 | <b>LC_ALL</b>                                   | If set to a non-empty string value, override the values of all the other                                                         |
| 26685 |                                                 | internationalization variables.                                                                                                  |
| 26686 | <b>LC_CTYPE</b>                                 | Determine the locale for the interpretation of sequences of bytes of text data as                                                |
| 26687 |                                                 | characters (for example, single-byte as opposed to multi-byte characters in                                                      |
| 26688 |                                                 | arguments).                                                                                                                      |
| 26689 | <b>LC_MESSAGES</b>                              |                                                                                                                                  |
| 26690 |                                                 | Determine the locale that should be used to affect the format and contents of                                                    |
| 26691 |                                                 | diagnostic messages written to standard error.                                                                                   |
| 26692 | XSI <b>NLSPATH</b>                              | Determine the location of message catalogs for the processing of <b>LC_MESSAGES</b> .                                            |
| 26693 | <b>PATH</b>                                     | Determine the search path used to locate the utility to be invoked. See the System                                               |
| 26694 |                                                 | Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 8, Environment                                                    |
| 26695 |                                                 | Variables.                                                                                                                       |
| 26696 | <b>ASYNCHRONOUS EVENTS</b>                      |                                                                                                                                  |
| 26697 |                                                 | Default.                                                                                                                         |
| 26698 | <b>STDOUT</b>                                   |                                                                                                                                  |
| 26699 |                                                 | Not used.                                                                                                                        |
| 26700 | <b>STDERR</b>                                   |                                                                                                                                  |
| 26701 |                                                 | Used only for diagnostic messages.                                                                                               |
| 26702 | <b>OUTPUT FILES</b>                             |                                                                                                                                  |
| 26703 |                                                 | None.                                                                                                                            |
| 26704 | <b>EXTENDED DESCRIPTION</b>                     |                                                                                                                                  |
| 26705 |                                                 | None.                                                                                                                            |
| 26706 | <b>EXIT STATUS</b>                              |                                                                                                                                  |
| 26707 |                                                 | If the <i>utility</i> utility is invoked, the exit status of <i>nice</i> shall be the exit status of <i>utility</i> ; otherwise, |
| 26708 |                                                 | the <i>nice</i> utility shall exit with one of the following values:                                                             |
| 26709 | 1-125                                           | An error occurred in the <i>nice</i> utility.                                                                                    |
| 26710 | 126                                             | The utility specified by <i>utility</i> was found but could not be invoked.                                                      |
| 26711 | 127                                             | The utility specified by <i>utility</i> could not be found.                                                                      |
| 26712 | <b>CONSEQUENCES OF ERRORS</b>                   |                                                                                                                                  |
| 26713 |                                                 | Default.                                                                                                                         |
| 26714 | <b>APPLICATION USAGE</b>                        |                                                                                                                                  |
| 26715 |                                                 | The only guaranteed portable uses of this utility are:                                                                           |
| 26716 | <i>nice utility</i>                             |                                                                                                                                  |
| 26717 |                                                 | Run <i>utility</i> with the default lower nice value.                                                                            |
| 26718 | <i>nice -n &lt;positive integer&gt; utility</i> |                                                                                                                                  |
| 26719 |                                                 | Run <i>utility</i> with a lower nice value.                                                                                      |
| 26720 |                                                 | On some systems they have no discernible effect on the invoked utility and on some others they                                   |
| 26721 |                                                 | are exactly equivalent.                                                                                                          |
| 26722 |                                                 | Historical systems have frequently supported the <i>&lt;positive integer&gt;</i> up to 20. Since there is no                     |
| 26723 |                                                 | error penalty associated with guessing a number that is too high, users without access to the                                    |
| 26724 |                                                 | system conformance document (to see what limits are actually in place) could use the historical                                  |
| 26725 |                                                 | 1 to 20 range or attempt to use very large numbers if the job should be truly low priority.                                      |

26726 The nice value of a process can be displayed using the command:

26727 `ps -o nice`

26728 The *command*, *env*, *nice*, *nohup*, *time*, and *xargs* utilities have been specified to use exit code 127 if  
26729 an error occurs so that applications can distinguish “failure to find a utility” from “invoked  
26730 utility exited with an error indication”. The value 127 was chosen because it is not commonly  
26731 used for other meanings; most utilities use small values for “normal error conditions” and the  
26732 values above 128 can be confused with termination due to receipt of a signal. The value 126 was  
26733 chosen in a similar manner to indicate that the utility could be found, but not invoked. Some  
26734 scripts produce meaningful error messages differentiating the 126 and 127 cases. The distinction  
26735 between exit codes 126 and 127 is based on KornShell practice that uses 127 when all attempts to  
26736 *exec* the utility fail with [ENOENT], and uses 126 when any attempt to *exec* the utility fails for  
26737 any other reason.

26738 Application writers should note that this utility need not be provided on systems that do not  
26739 support the User Portability Utilities option.

#### 26740 EXAMPLES

26741 None.

#### 26742 RATIONALE

26743 Due to the text about the limits of the nice value being implementation-dependent, *nice* is not  
26744 actually required to change the nice value of the executed command; the limits could be zero  
26745 differences from the system default, although the implementor is required to document this fact  
26746 in the conformance document.

26747 The 4.3 BSD version of *nice* does not check if *increment* is a valid decimal integer. The command  
26748 *nice -x utility*, for example, would be treated the same as the command *nice --1 utility*. If the  
26749 user does not have appropriate privileges, this results in a “permission denied” error. This is  
26750 considered a bug.

26751 When a user without appropriate privileges gives a negative *increment*, System V treats it like  
26752 the command *nice -0 utility*, while 4.3 BSD writes a “permission denied” message and does not  
26753 run the utility. Neither was considered clearly superior, so the behavior was left unspecified.

26754 The C shell has a built-in version of *nice* that has a different interface from the one described in  
26755 this volume of IEEE Std. 1003.1-200x.

26756 The term “utility” is used, rather than “command”, to highlight the fact that shell compound  
26757 commands, pipelines, and so on, cannot be used. Special built-ins also cannot be used.  
26758 However, “utility” includes user application programs and shell scripts, not just utilities defined  
26759 in this volume of IEEE Std. 1003.1-200x.

26760 Historical implementations of *nice* provide a nice value range of 40 or 41 discrete steps, with the  
26761 default nice value being the midpoint of that range. By default, they lower the nice value of the  
26762 executed utility by 10.

26763 Some historical documentation states that the *increment* value must be within a fixed range. This  
26764 is misleading; the valid *increment* values on any invocation are determined by the current  
26765 process nice value, which is not always the default.

26766 The definition of nice value is not intended to suggest that all processes in a system have  
26767 priorities that are comparable. Scheduling policy extensions such as the realtime priorities in  
26768 POSIX.4 make the notion of a single underlying priority for all scheduling policies problematic.  
26769 Some systems may implement the *nice* **-related** features to affect all processes on the system,  
26770 others to affect just the general time-sharing activities implied by this volume of  
26771 IEEE Std. 1003.1-200x, and others may have no effect at all. Because of the use of



26772 “implementation-dependent” in *nice* and *renice*, a wide range of implementation strategies are  
26773 possible.

26774 **FUTURE DIRECTIONS**

26775 None.

26776 **SEE ALSO**

26777 *renice*

26778 **CHANGE HISTORY**

26779 First released in Issue 4.

26780 **Issue 6**

26781 This utility is now marked as part of the User Portability Utilities option.

26782 The obsolescent SYNOPSIS is removed.

## 26783 NAME

26784 nl — line numbering filter

## 26785 SYNOPSIS

```
26786 xsi nl [-p][-b type][-d delim][-f type][-h type][-i incr][-l num][-n format]
26787 [-s sep][-v startnum][-w width][file]
26788
```

## 26789 DESCRIPTION

26790 The *nl* utility shall read lines from the named *file* or the standard input if no *file* is named and  
 26791 shall reproduce the lines to standard output. Lines shall be numbered on the left. Additional  
 26792 functionality may be provided in accordance with the command options in effect.

26793 The *nl* utility views the text it reads in terms of logical pages. Line numbering is reset at the start  
 26794 of each logical page. A logical page consists of a header, a body, and a footer section. Empty  
 26795 sections are valid. Different line numbering options are independently available for header,  
 26796 body, and footer (for example, no numbering of header and footer lines while numbering blank  
 26797 lines only in the body).

26798 The starts of logical page sections are signaled by input lines containing nothing but the  
 26799 following delimiter characters:

26800

26801

26802

26803

| Line     | Start of |
|----------|----------|
| \: \: \: | Header   |
| \: \:    | Body     |
| \:       | Footer   |

26804 Unless otherwise specified, *nl* assumes the text being read is in a single logical page body.

## 26805 OPTIONS

26806 The *nl* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 26807 Section 12.2, Utility Syntax Guidelines. Only one file can be named.

26808 The following options shall be supported:

26809 **-b type** Specify which logical page body lines shall be numbered. Recognized *types* and  
 26810 their meaning are:

26811 **a** Number all lines.

26812 **t** Number only non-empty lines.

26813 **n** No line numbering.

26814 **pstring** Number only lines that contain the basic regular expression specified in  
 26815 *string*.

26816 The default *type* for logical page body is **t** (text lines numbered).

26817 **-d delim** Specify the delimiter characters that indicate the start of a logical page section.  
 26818 These can be changed from the default characters "\:" to two user-specified  
 26819 characters. If only one character is entered, the second character remains the  
 26820 default character ' '.

26821 **-f type** Specify the same as **b type** except for footer. The default for logical page footer is **n**  
 26822 (no lines numbered).

26823 **-h type** Specify the same as **b type** except for header. The default *type* for logical page  
 26824 header is **n** (no lines numbered).

|       |                    |                                                                                                                                                                                                                                                                                   |
|-------|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 26825 | <b>-i incr</b>     | Specify the increment value used to number logical page lines. The default is 1.                                                                                                                                                                                                  |
| 26826 | <b>-l num</b>      | Specify the number of blank lines to be considered as one. For example, <b>-l 2</b> results in only the second adjacent blank line being numbered (if the appropriate <b>-h a</b> , <b>-b a</b> , or <b>-f a</b> option is set). The default is 1.                                |
| 26827 |                    |                                                                                                                                                                                                                                                                                   |
| 26828 |                    |                                                                                                                                                                                                                                                                                   |
| 26829 | <b>-n format</b>   | Specify the line numbering format. Recognized values are: <b>ln</b> , left justified, leading zeros suppressed; <b>rn</b> , right justified, leading zeros suppressed; <b>rz</b> , right justified, leading zeros kept. The default <i>format</i> is <b>rn</b> (right justified). |
| 26830 |                    |                                                                                                                                                                                                                                                                                   |
| 26831 |                    |                                                                                                                                                                                                                                                                                   |
| 26832 | <b>-p</b>          | Specify that numbering should not be restarted at logical page delimiters.                                                                                                                                                                                                        |
| 26833 | <b>-s sep</b>      | Specify the characters used in separating the line number and the corresponding text line. The default <i>sep</i> is a <tab>.                                                                                                                                                     |
| 26834 |                    |                                                                                                                                                                                                                                                                                   |
| 26835 | <b>-v startnum</b> | Specify the initial value used to number logical page lines. The default is 1.                                                                                                                                                                                                    |
| 26836 | <b>-w width</b>    | Specify the number of characters to be used for the line number. The default <i>width</i> is 6.                                                                                                                                                                                   |
| 26837 |                    |                                                                                                                                                                                                                                                                                   |

#### 26838 OPERANDS

26839 The following operand shall be supported:

26840 *file* A path name of a text file to be line-numbered.

#### 26841 STDIN

26842 The standard input is a text file that is used if no *file* operand is given.

#### 26843 INPUT FILES

26844 The input file named by the *file* operand is a text file.

#### 26845 ENVIRONMENT VARIABLES

26846 The following environment variables shall affect the execution of *nl*:

26847 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 26848 If *LANG* is unset or null, the corresponding value from the implementation-  
 26849 dependent default locale shall be used. If any of the internationalization variables  
 26850 contains an invalid setting, the utility shall behave as if none of the variables had  
 26851 been defined.

26852 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 26853 internationalization variables.

26854 *LC\_COLLATE*  
 26855 Determine the locale for the behavior of ranges, equivalence classes and multi-  
 26856 character collating elements within regular expressions.

26857 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 26858 characters (for example, single-byte as opposed to multi-byte characters in  
 26859 arguments and input files), the behavior of character classes within regular  
 26860 expressions, and for deciding which characters are in character class **graph** (for the  
 26861 **-b t**, **-f t**, and **-h t** options).

26862 *LC\_MESSAGES*  
 26863 Determine the locale that should be used to affect the format and contents of  
 26864 diagnostic messages written to standard error.

26865 *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

**26866 ASYNCHRONOUS EVENTS**

26867 Default.

**26868 STDOUT**

26869 The standard output shall be a text file in the following format:

26870 "%s%s%s", <line number>, <separator>, <input line>

26871 where <line number> is one of the following numeric formats:

26872 %6d When the **rn** format is used (the default; see **-n**).

26873 %06d When the **rz** format is used.

26874 %-6d When the **ln** format is used.

26875 <empty> When line numbers are suppressed for a portion of the page; the <separator> is also  
26876 suppressed.

26877 In the preceding list, the number 6 is the default width; the **-w** option can change this value.

**26878 STDERR**

26879 Used only for diagnostic messages.

**26880 OUTPUT FILES**

26881 None.

**26882 EXTENDED DESCRIPTION**

26883 None.

**26884 EXIT STATUS**

26885 The following exit values shall be returned:

26886 0 Successful completion.

26887 >0 An error occurred.

**26888 CONSEQUENCES OF ERRORS**

26889 Default.

**26890 APPLICATION USAGE**

26891 In using the **-d delim** option, care should be taken to escape characters that have special meaning  
26892 to the command interpreter.

**26893 EXAMPLES**

26894 The command:

26895 nl -v 10 -i 10 -d \!+ file1

26896 numbers *file1* starting at line number 10 with an increment of 10. The logical page delimiter is  
26897 " !+ ". Note that the ' ! ' has to be escaped when using *csh* as a command interpreter because of  
26898 its history substitution syntax. For *ksh* and *sh* the escape is not necessary, but does not do any  
26899 harm.

**26900 RATIONALE**

26901 None.

**26902 FUTURE DIRECTIONS**

26903 The intermingling of the *file* operand with the options is an obsolescent feature that is removed  
26904 from a future issue.

26905 **SEE ALSO**26906 *pr*26907 **CHANGE HISTORY**

26908 First released in Issue 2.

26909 **Issue 4**

26910 Format reorganized.

26911 Utility Syntax Guideline support mandated.

26912 Internationalized environment variable support mandated.

26913 **Issue 5**26914 The option `[-f type]` is added to the SYNOPSIS. The option descriptions are presented in  
26915 alphabetic order. The description of `-bt` is changed to “Number only non-empty lines”.26916 **Issue 6**26917 The obsolescent behavior allowing the options to be intermingled with the optional *file* operand  
26918 is removed.

26919 **NAME**26920 nm — write the name list of an object file (**DEVELOPMENT**)26921 **SYNOPSIS**26922 UP SD XSI nm [-APv][-efox][-g|-u][-t *format*] *file...*

26923

26924 **DESCRIPTION**

26925 This utility shall be provided on systems that support both the User Portability Utilities option  
 26926 and the Software Development Utilities option. On other systems it is optional. Certain options  
 26927 are only available on XSI-conformant systems.

26928 The *nm* utility shall display symbolic information appearing in the object file, executable file or  
 26929 object-file library named by *file*. If no symbolic information is available for a valid input file, the  
 26930 *nm* utility shall report that fact, but not consider it an error condition.

26931 XSI The default base used when numeric values are written is **decimal**.

26932 **OPTIONS**

26933 The *nm* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 26934 Section 12.2, Utility Syntax Guidelines.

26935 The following options shall be supported:

26936 **-A** Write the full path name or library name of an object on each line.

26937 XSI **-e** Write only external (global) and static symbol information.

26938 XSI **-f** Produce full output. Write redundant symbols (**.text**, **.data**, and **.bss**), normally  
 26939 suppressed.

26940 **-g** Write only external (global) symbol information.

26941 XSI **-o** Write numeric values in octal (equivalent to **-t o**).

26942 **-P** Write information in a portable output format, as specified in the STDOUT section.

26943 **-t format** Write each numeric value in the specified format. The format shall be dependent  
 26944 on the single character used as the *format* option-argument:

26945 XSI d The offset is written in decimal (**default**).

26946 o The offset is written in octal.

26947 x The offset is written in hexadecimal.

26948 **-u** Write only undefined symbols.

26949 **-v** Sort output by value instead of alphabetically.

26950 XSI **-x** Write numeric values in hexadecimal (equivalent to **-t x**).

26951 **OPERANDS**

26952 The following operand shall be supported:

26953 *file* A path name of an object file, executable file, or object-file library.

26954 **STDIN**

26955 See the INPUT FILES section.

26956 **INPUT FILES**

26957       The input file shall be an object file, an object-file library whose format is the same as those  
 26958       produced by the *ar* utility for link editing, or an executable file. The *nm* utility may accept  
 26959       additional implementation-dependent object library formats for the input file.

26960 **ENVIRONMENT VARIABLES**

26961       The following environment variables shall affect the execution of *nm*:

26962       *LANG*       Provide a default value for the internationalization variables that are unset or null.  
 26963       If *LANG* is unset or null, the corresponding value from the implementation-  
 26964       dependent default locale shall be used. If any of the internationalization variables  
 26965       contains an invalid setting, the utility shall behave as if none of the variables had  
 26966       been defined.

26967       *LC\_ALL*     If set to a non-empty string value, override the values of all the other  
 26968       internationalization variables.

26969       *LC\_COLLATE* Determine the locale for character collation information for the symbol-name and  
 26970       symbol-value collation sequences.  
 26971

26972       *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 26973       characters (for example, single-byte as opposed to multi-byte characters in  
 26974       arguments).

26975       *LC\_MESSAGES* Determine the locale that should be used to affect the format and contents of  
 26976       diagnostic messages written to standard error.  
 26977

26978 XSI       *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

26979 **ASYNCHRONOUS EVENTS**

26980       Default.

26981 **STDOUT**

26982       If symbolic information is present in the input files, then for each file or for each member of an  
 26983       archive, the *nm* utility shall write the following information to standard output. By default, the  
 26984       format is unspecified, but the output shall be sorted alphabetically by symbol name:

- 26985       • Library or object name, if *-A* is specified
- 26986       • Symbol name
- 26987       • Symbol type, which shall either be one of the following single characters or an  
 26988       implementation-dependent type represented by a single character:
- 26989       A   Global absolute symbol.
- 26990       a   Local absolute symbol.
- 26991       B   Global “bss” (that is, uninitialized data space) symbol.
- 26992       b   Local bss symbol.
- 26993       D   Global data symbol.
- 26994       d   Local data symbol.
- 26995       T   Global text symbol.
- 26996       t   Local text symbol.

26997           U    Undefined symbol.

26998           • Value of the symbol

26999           • The size associated with the symbol, if applicable

27000       This information may be supplemented by additional information specific to the  
27001       implementation.

27002       If the **-P** option is specified, the previous information shall be displayed using the following  
27003       portable format. The three versions differ depending on whether **-t d**, **-t o**, or **-t x** was specified,  
27004       respectively:

27005       "%s%s %s %d %d\n", <library/object name>, <name>, <type>,  
27006       <value>, <size>

27007       "%s%s %s %o %o\n", <library/object name>, <name>, <type>,  
27008       <value>, <size>

27009       "%s%s %s %x %x\n", <library/object name>, <name>, <type>,  
27010       <value>, <size>

27011       where

27012       <library/object name> shall be formatted as follows:

27013       • If **-A** is not specified, <library/object name> shall be an empty string.

27014       • If **-A** is specified and the corresponding *file* operand does not name a library:

27015       "%s: ", <file>

27016       • If **-A** is specified and the corresponding *file* operand names a library. In this case, <object file>  
27017       shall name the object file in the library containing the symbol being described:

27018       "%s[%s]: ", <file>, <object file>

27019       If **-A** is not specified, then if more than one *file* operand is specified or if only one *file* operand is  
27020       specified and it names a library, *nm* shall write a line identifying the object containing the  
27021       following symbols before the lines containing those symbols, in the form:

27022       • If the corresponding *file* operand does not name a library:

27023       "%s:\n", <file>

27024       • If the corresponding *file* operand names a library; in this case, <object file> shall be the name  
27025       of the file in the library containing the following symbols:

27026       "%s[%s]:\n", <file>, <object file>

27027       If **-P** is specified, but **-t** is not, the format shall be as if **-t x** had been specified.

27028   **STDERR**

27029       Used only for diagnostic messages.

27030   **OUTPUT FILES**

27031       None.

27032   **EXTENDED DESCRIPTION**

27033       None.



**27034 EXIT STATUS**

27035           The following exit values shall be returned:

27036           0   Successful completion.

27037           >0   An error occurred.

**27038 CONSEQUENCES OF ERRORS**

27039           Default.

**27040 APPLICATION USAGE**

27041           Mechanisms for dynamic linking make this utility less meaningful when applied to an  
27042           executable file because a dynamically linked executable may omit numerous library routines  
27043           that would be found in a statically linked executable.

**27044 EXAMPLES**

27045           None.

**27046 RATIONALE**

27047           Historical implementations of *nm* have used different bases for numeric output and supplied  
27048           different default types of symbols that were reported. The *-t format* option, similar to that used  
27049           in *od* and *strings*, can be used to specify the numeric base; *-g* and *-u* can be used to restrict the  
27050           amount of output or the types of symbols included in the output.

27051           The option list was significantly reduced from that provided by historical implementations.

27052           The *nm* description is a subset of both the System V and BSD *nm* utilities with no specified  
27053           default output.

27054           It was recognized that mechanisms for dynamic linking make this utility less meaningful when  
27055           applied to an executable file (because a dynamically linked executable file may omit numerous  
27056           library routines that would be found in a statically linked executable file), but the value of *nm*  
27057           during software development was judged to outweigh other limitations.

27058           The compromise of using *-t format* versus using *-d*, *-o*, and other similar options was necessary  
27059           because of differences in the meaning of *-o* between implementations. The *-o* option from BSD  
27060           has been provided here as *-A* to avoid confusion with the *-o* from System V (which has been  
27061           provided here as *-t* and as *-o* on XSI-conformant systems).

27062           The default output format of *nm* is not specified because of differences in historical  
27063           implementations. The *-P* option was added to allow some type of portable output format. After  
27064           a comparison of the different formats used in SunOS, BSD, SVR3, and SVR4, it was decided to  
27065           create one that did not match the current format of any of these four systems. The format  
27066           devised is easy to parse by humans, easy to parse in shell scripts, and does not need to vary  
27067           depending on locale (because no English descriptions are included). All of the systems currently  
27068           have the information available to use this format.

27069           The format given in *nm* STDOUT uses spaces between the fields, which may be any number of  
27070           <blank>s required to align the columns. The single-character types were selected to match  
27071           historical practice, and the requirement that implementation additions also be single characters  
27072           made parsing the information easier for shell scripts.

**27073 FUTURE DIRECTIONS**

27074           None.

**27075 SEE ALSO**

27076           *ar*, *c89*

27077 **CHANGE HISTORY**

27078 First released in Issue 2.

27079 **Issue 4**

27080 Aligned with the ISO/IEC 9945-2: 1993 standard.

27081 **Issue 6**27082 This utility is now marked as supported when both the User Portability Utilities option and the  
27083 Software Development Utilities option are supported.

27084 **NAME**

27085           nohup — invoke a utility immune to hangups

27086 **SYNOPSIS**27087           nohup *utility* [*argument...*]27088 **DESCRIPTION**

27089           The *nohup* utility shall invoke the utility named by the *utility* operand with arguments supplied  
 27090           as the *argument* operands. At the time the named *utility* is invoked, the SIGHUP signal shall be  
 27091           set to be ignored.

27092           If the standard output is a terminal, all output written by the named *utility* to its standard output  
 27093           shall be appended to the end of the file **nohup.out** in the current directory. If **nohup.out** cannot  
 27094           be created or opened for appending, the output shall be appended to the end of the file  
 27095           **nohup.out** in the directory specified by the *HOME* environment variable. If neither file can be  
 27096           created or opened for appending, *utility* shall not be invoked. If a file is created, the file's  
 27097           permission bits shall be set to S\_IRUSR | S\_IWUSR.

27098           If the standard error is a terminal, all output written by the named *utility* to its standard error  
 27099           shall be redirected to the same file descriptor as the standard output.

27100 **OPTIONS**

27101           None.

27102 **OPERANDS**

27103           The following operands shall be supported:

27104           *utility*       The name of a utility that is to be invoked. If the *utility* operand names any of the  
 27105           special built-in utilities in Section 2.14 on page 96, the results are undefined.

27106           *argument*   Any string to be supplied as an argument when invoking the utility named by the  
 27107           *utility* operand.

27108 **STDIN**

27109           Not used.

27110 **INPUT FILES**

27111           None.

27112 **ENVIRONMENT VARIABLES**27113           The following environment variables shall affect the execution of *nohup*:

27114           *HOME*       Determine the path name of the user's home directory; if the output file **nohup.out**  
 27115           cannot be created in the current directory, the *nohup* utility shall use the directory  
 27116           named by *HOME* to create the file.

27117           *LANG*       Provide a default value for the internationalization variables that are unset or null.  
 27118           If *LANG* is unset or null, the corresponding value from the implementation-  
 27119           dependent default locale shall be used. If any of the internationalization variables  
 27120           contains an invalid setting, the utility behave as if none of the variables had been  
 27121           defined.

27122           *LC\_ALL*      If set to a non-empty string value, override the values of all the other  
 27123           internationalization variables.

27124           *LC\_CTYPE*   Determine the locale for the interpretation of sequences of bytes of text data as  
 27125           characters (for example, single-byte as opposed to multi-byte characters in  
 27126           arguments).

|       |                               |                                                                                                                                                     |
|-------|-------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| 27127 | <i>LC_MESSAGES</i>            |                                                                                                                                                     |
| 27128 |                               | Determine the locale that should be used to affect the format and contents of                                                                       |
| 27129 |                               | diagnostic messages written to standard error.                                                                                                      |
| 27130 | XSI <i>NLSPATH</i>            | Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .                                                               |
| 27131 | <i>PATH</i>                   | Determine the search path that is used to locate the utility to be invoked. See the                                                                 |
| 27132 |                               | System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 8,                                                                            |
| 27133 |                               | Environment Variables.                                                                                                                              |
| 27134 | <b>ASYNCHRONOUS EVENTS</b>    |                                                                                                                                                     |
| 27135 |                               | The <i>nohup</i> utility shall take the standard action for all signals except that SIGHUP shall be                                                 |
| 27136 |                               | ignored.                                                                                                                                            |
| 27137 | <b>STDOUT</b>                 |                                                                                                                                                     |
| 27138 |                               | If the standard output is not a terminal, the standard output of <i>nohup</i> shall be the standard                                                 |
| 27139 |                               | output generated by the execution of the <i>utility</i> specified by the operands. Otherwise, nothing                                               |
| 27140 |                               | shall be written to the standard output.                                                                                                            |
| 27141 | <b>STDERR</b>                 |                                                                                                                                                     |
| 27142 |                               | If the standard output is a terminal, a message shall be written to the standard error, indicating                                                  |
| 27143 |                               | the name of the file to which the output is being appended. The name of the file shall be either                                                    |
| 27144 |                               | <b>nohup.out</b> or <b>\$HOME/nohup.out</b> .                                                                                                       |
| 27145 | <b>OUTPUT FILES</b>           |                                                                                                                                                     |
| 27146 |                               | If the standard output is a terminal, all output written by the named <i>utility</i> to the standard                                                |
| 27147 |                               | output and standard error is appended to the file <b>nohup.out</b> , which is created if it does not                                                |
| 27148 |                               | already exist.                                                                                                                                      |
| 27149 | <b>EXTENDED DESCRIPTION</b>   |                                                                                                                                                     |
| 27150 |                               | None.                                                                                                                                               |
| 27151 | <b>EXIT STATUS</b>            |                                                                                                                                                     |
| 27152 |                               | The following exit values shall be returned:                                                                                                        |
| 27153 | 126                           | The utility specified by <i>utility</i> was found but could not be invoked.                                                                         |
| 27154 | 127                           | An error occurred in the <i>nohup</i> utility or the utility specified by <i>utility</i> could not be                                               |
| 27155 |                               | found.                                                                                                                                              |
| 27156 |                               | Otherwise, the exit status of <i>nohup</i> shall be that of the utility specified by the <i>utility</i> operand.                                    |
| 27157 | <b>CONSEQUENCES OF ERRORS</b> |                                                                                                                                                     |
| 27158 |                               | Default.                                                                                                                                            |
| 27159 | <b>APPLICATION USAGE</b>      |                                                                                                                                                     |
| 27160 |                               | The <i>command</i> , <i>env</i> , <i>nice</i> , <i>nohup</i> , <i>time</i> , and <i>xargs</i> utilities have been specified to use exit code 127 if |
| 27161 |                               | an error occurs so that applications can distinguish “failure to find a utility” from “invoked                                                      |
| 27162 |                               | utility exited with an error indication”. The value 127 was chosen because it is not commonly                                                       |
| 27163 |                               | used for other meanings; most utilities use small values for “normal error conditions” and the                                                      |
| 27164 |                               | values above 128 can be confused with termination due to receipt of a signal. The value 126 was                                                     |
| 27165 |                               | chosen in a similar manner to indicate that the utility could be found, but not invoked. Some                                                       |
| 27166 |                               | scripts produce meaningful error messages differentiating the 126 and 127 cases. The distinction                                                    |
| 27167 |                               | between exit codes 126 and 127 is based on KornShell practice that uses 127 when all attempts to                                                    |
| 27168 |                               | <i>exec</i> the utility fail with [ENOENT], and uses 126 when any attempt to <i>exec</i> the utility fails for                                      |
| 27169 |                               | any other reason.                                                                                                                                   |

27170 **EXAMPLES**

27171 It is frequently desirable to apply *nohup* to pipelines or lists of commands. This can be done by  
27172 placing pipelines and command lists in a single file; this file can then be invoked as a utility, and  
27173 the *nohup* applies to everything in the file.

27174 Alternatively, the following command can be used to apply *nohup* to a complex command:

27175 `nohup sh -c 'complex-command-line'`

27176 **RATIONALE**

27177 The 4.3 BSD version ignores SIGTERM and SIGHUP, and if `./nohup.out` cannot be used, it fails  
27178 instead of trying to use `$HOME/nohup.out`.

27179 The *cs*h utility has a built-in version of *nohup* that acts differently from the POSIX Shell and  
27180 Utilities *nohup*.

27181 The term *utility* is used, rather than *command*, to highlight the fact that shell compound  
27182 commands, pipelines, special built-ins, and so on, cannot be used directly. However, *utility*  
27183 includes user application programs and shell scripts, not just the standard utilities.

27184 Historical versions of the *nohup* utility use default file creation semantics. Some more recent  
27185 versions use the permissions specified here as an added security precaution.

27186 Some historical implementations ignore SIGQUIT in addition to SIGHUP; others ignore  
27187 SIGTERM. An early proposal allowed, but did not require, SIGQUIT to be ignored. Several  
27188 reviewers objected that *nohup* should only modify the handling of SIGHUP as required by this  
27189 volume of IEEE Std. 1003.1-200x.

27190 **FUTURE DIRECTIONS**

27191 None.

27192 **SEE ALSO**

27193 *sh*, the System Interfaces volume of IEEE Std. 1003.1-200x, *signal()*

27194 **CHANGE HISTORY**

27195 First released in Issue 2.

27196 **Issue 4**

27197 Aligned with the ISO/IEC 9945-2:1993 standard.

27198 **NAME**

27199       od — dump files in various formats

27200 **SYNOPSIS**

27201       od [-v][-A *address\_base*][-j *skip*][-N *count*][-t *type\_string*]...  
 27202       [*file*...]

27203 XSI       od [-bcdosx][*file*] [[+]offset[.][b]]

27204

27205 **DESCRIPTION**

27206       The *od* utility shall write the contents of its input files to standard output in a user-specified  
 27207       format.

27208 **OPTIONS**

27209       The *od* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 27210 XSI       Section 12.2, Utility Syntax Guidelines, except that the order of presentation of the **-t** options  
 27211       and the **-bcdosx** options is significant.

27212       The following options shall be supported:

27213       **-A** *address\_base*

27214               Specify the input offset base. See the EXTENDED DESCRIPTION section. The  
 27215               application shall ensure that the *address\_base* option-argument is a character. The  
 27216               characters 'd', 'o', and 'x' specify that the offset base shall be written in  
 27217               decimal, octal, or hexadecimal, respectively. The character 'n' specifies that the  
 27218               offset shall not be written.

27219 XSI       **-b**       Interpret bytes in octal. This is equivalent to **-t o1**.

27220 XSI       **-c**       Interpret bytes as characters specified by the current setting of the *LC\_CTYPE*  
 27221               category. Certain non-graphic characters appear as C escapes: "NUL=\0",  
 27222               "BS=\b", "FF=\f", "NL=\n", "CR=\r", "HT=\t"; others appear as 3-digit octal  
 27223               numbers.

27224 XSI       **-d**       Interpret *words* (two-byte units) in unsigned decimal. This is equivalent to **-t u2**.

27225       **-j** *skip*       Jump over *skip* bytes from the beginning of the input. The *od* utility shall read or  
 27226               seek past the first *skip* bytes in the concatenated input files. If the combined input  
 27227               is not at least *skip* bytes long, the *od* utility shall write a diagnostic message to  
 27228               standard error and exit with a non-zero exit status.

27229               By default, the *skip* option-argument shall be interpreted as a decimal number.  
 27230               With a leading "0x" or "0X", the offset shall be interpreted as a hexadecimal  
 27231               number; otherwise, with a leading '0', the offset shall be interpreted as an octal  
 27232               number. Appending the character 'b', 'k', or 'm' to offset shall cause it to be  
 27233 XSI               interpreted as a multiple of 512, 1024, or 1048576 bytes, respectively. If the *skip*  
 27234               number is hexadecimal, any appended 'b' shall be considered to be the final  
 27235               hexadecimal digit.

27236       **-N** *count*       Format no more than *count* bytes of input. By default, *count* shall be interpreted as  
 27237               a decimal number. With a leading "0x" or "0X", *count* shall be interpreted as a  
 27238               hexadecimal number; otherwise, with a leading '0', it shall be interpreted as an  
 27239               octal number. If *count* bytes of input (after successfully skipping, if **-j** *skip* is  
 27240               specified) are not available, it shall not be considered an error; the *od* utility shall  
 27241               format the input that is available.

27242 XSI       **-o**       Interpret *words* (two-byte units) in octal. This is equivalent to **-t o2**.

|       |     |                                      |                                                                                                                                            |
|-------|-----|--------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------|
| 27243 | XSI | <b>-s</b>                            | Interpret <i>words</i> (two-byte units) in signed decimal. This is equivalent to <b>-t d2</b> .                                            |
| 27244 |     | <b>-t <i>type_string</i></b>         |                                                                                                                                            |
| 27245 |     |                                      | Specify one or more output types. See the EXTENDED DESCRIPTION section. The                                                                |
| 27246 |     |                                      | application shall ensure that the <i>type_string</i> option-argument is a string specifying                                                |
| 27247 |     |                                      | the types to be used when writing the input data. The string shall consist of the                                                          |
| 27248 |     |                                      | type specification characters <b>a</b> , <b>c</b> , <b>d</b> , <b>f</b> , <b>o</b> , <b>u</b> , and <b>x</b> , specifying named character, |
| 27249 |     |                                      | character, signed decimal, floating point, octal, unsigned decimal, and                                                                    |
| 27250 |     |                                      | hexadecimal, respectively. The type specification characters <b>d</b> , <b>f</b> , <b>o</b> , <b>u</b> , and <b>x</b> can be               |
| 27251 |     |                                      | followed by an optional unsigned decimal integer that specifies the number of                                                              |
| 27252 |     |                                      | bytes to be transformed by each instance of the output type. The type specification                                                        |
| 27253 |     |                                      | character <b>f</b> can be followed by an optional <b>F</b> , <b>D</b> , or <b>L</b> indicating that the conversion                         |
| 27254 |     |                                      | should be applied to an item of type <b>float</b> , <b>double</b> , or <b>long double</b> , respectively.                                  |
| 27255 |     |                                      | The type specification characters <b>d</b> , <b>o</b> , <b>u</b> and <b>x</b> can be followed by an optional <b>C</b> , <b>S</b> ,         |
| 27256 |     |                                      | <b>I</b> , or <b>L</b> indicating that the conversion should be applied to an item of type <b>char</b> ,                                   |
| 27257 |     |                                      | <b>short</b> , <b>int</b> , or <b>long</b> , respectively. Multiple types can be concatenated within the                                   |
| 27258 |     |                                      | same <i>type_string</i> and multiple <b>-t</b> options can be specified. Output lines shall be                                             |
| 27259 |     |                                      | written for each type specified in the order in which the type specification                                                               |
| 27260 |     |                                      | characters are specified.                                                                                                                  |
| 27261 |     | <b>-v</b>                            | Write all input data. Without the <b>-v</b> option, any number of groups of output lines,                                                  |
| 27262 |     |                                      | which would be identical to the immediately preceding group of output lines                                                                |
| 27263 |     |                                      | (except for the byte offsets), shall be replaced with a line containing only an                                                            |
| 27264 |     |                                      | asterisk ( <b>' * '</b> ).                                                                                                                 |
| 27265 | XSI | <b>-x</b>                            | Interpret <i>words</i> (two-byte units) in hexadecimal. This is equivalent to <b>-t x2</b> .                                               |
| 27266 | XSI |                                      | Multiple types can be specified by using multiple <b>-bcdostx</b> options. Output lines are written for                                    |
| 27267 |     |                                      | each type specified in the order in which the types are specified.                                                                         |
| 27268 |     | <b>OPERANDS</b>                      |                                                                                                                                            |
| 27269 |     |                                      | The following operands shall be supported:                                                                                                 |
| 27270 |     | <b>file</b>                          | A path name of a file to be read. If no file operands are specified, the standard                                                          |
| 27271 |     |                                      | input shall be used. If the first character of <i>file</i> is a plus sign ( <b>' + '</b> ) or the first                                    |
| 27272 |     |                                      | character of the first file operand is numeric, no more than two operands are given,                                                       |
| 27273 | XSI |                                      | and none of the <b>-A</b> , <b>-j</b> , <b>-N</b> , or <b>-t</b> options is specified, the operand is assumed to be                        |
| 27274 |     |                                      | an <i>offset</i> .                                                                                                                         |
| 27275 | XSI | <b>[+]<i>offset</i>[.][<i>b</i>]</b> |                                                                                                                                            |
| 27276 |     |                                      | The <i>offset</i> operand specifies the offset in the file where dumping is to commence.                                                   |
| 27277 |     |                                      | This operand is normally interpreted as octal bytes. If <b>' . '</b> is appended, the offset                                               |
| 27278 |     |                                      | shall be interpreted in decimal. If <b>' b '</b> is appended, the offset shall be interpreted                                              |
| 27279 |     |                                      | in units of 512 bytes. If the <i>file</i> argument is omitted, and none of the <b>-A</b> , <b>-j</b> , <b>-N</b> , or                      |
| 27280 |     |                                      | <b>-t</b> options is specified, the application shall ensure that the offset argument is                                                   |
| 27281 |     |                                      | preceded by <b>' + '</b> .                                                                                                                 |
| 27282 |     | <b>STDIN</b>                         |                                                                                                                                            |
| 27283 |     |                                      | The standard input shall be used only if no <i>file</i> operands are specified. See the INPUT FILES                                        |
| 27284 |     |                                      | section.                                                                                                                                   |
| 27285 |     | <b>INPUT FILES</b>                   |                                                                                                                                            |
| 27286 |     |                                      | The input files can be any file type.                                                                                                      |

27287 **ENVIRONMENT VARIABLES**

27288 The following environment variables shall affect the execution of *od*:

27289 **LANG** Provide a default value for the internationalization variables that are unset or null.  
 27290 If *LANG* is unset or null, the corresponding value from the implementation-  
 27291 dependent default locale shall be used. If any of the internationalization variables  
 27292 contains an invalid setting, the utility shall behave as if none of the variables had  
 27293 been defined.

27294 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 27295 internationalization variables.

27296 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 27297 characters (for example, single-byte as opposed to multi-byte characters in  
 27298 arguments and input files).

27299 **LC\_MESSAGES**  
 27300 Determine the locale that should be used to affect the format and contents of  
 27301 diagnostic messages written to standard error.

27302 **LC\_NUMERIC**  
 27303 Determine the locale for selecting the radix character used when writing floating-  
 27304 point formatted output.

27305 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

27306 **ASYNCHRONOUS EVENTS**

27307 Default.

27308 **STDOUT**

27309 See the EXTENDED DESCRIPTION section.

27310 **STDERR**

27311 Used only for diagnostic messages.

27312 **OUTPUT FILES**

27313 None.

27314 **EXTENDED DESCRIPTION**

27315 The *od* utility shall copy sequentially each input file to standard output, transforming the input  
 27316 XSI data according to the output types specified by the **-t** options or the **-bcdosx** options. If no  
 27317 output type is specified, the default output shall be as if **-t oS** had been specified.

27318 The number of bytes transformed by the output type specifier *c* may be variable depending on  
 27319 the *LC\_CTYPE* category.

27320 The default number of bytes transformed by output type specifiers **d**, **f**, **o**, **u**, and **x** corresponds  
 27321 to the various C-language types as follows. If the *c89* compiler is present on the system, these  
 27322 specifiers shall correspond to the sizes used by default in that compiler. Otherwise, these sizes  
 27323 may vary among systems that conform to IEEE Std. 1003.1-200x.

- 27324 • For the type specifier characters **d**, **o**, **u**, and **x**, the default number of bytes shall correspond
- 27325 to the size of the underlying implementation's basic integral data type. For these specifier
- 27326 characters, the implementation shall support values of the optional number of bytes to be
- 27327 converted corresponding to the number of bytes in the C-language types **char**, **short**, **int**, and
- 27328 **long**. These numbers can also be specified by an application as the characters **'C'**, **'S'**, **'I'**,
- 27329 and **'L'**, respectively. The implementation shall also support the values 1, 2, and 4, even if it
- 27330 provides no C-Language types of those sizes. The byte order used when interpreting numeric
- 27331 values is implementation-dependent, but shall correspond to the order in which a constant of



the corresponding type is stored in memory on the system.

- For the type specifier character **f**, the default number of bytes shall correspond to the number of bytes in the underlying implementation's basic double precision floating-point data type. The implementation shall support values of the optional number of bytes to be converted corresponding to the number of bytes in the C-language types **float**, **double**, and **long double**. These numbers can also be specified by an application as the characters '**F**', '**D**', and '**L**', respectively.

The type specifier character **a** specifies that bytes are interpreted as named characters from the International Reference Version (IRV) of the ISO/IEC 646:1991 standard. Only the least significant seven bits of each byte shall be used for this type specification. Bytes with the values listed in the following table shall be written using the corresponding names for those characters.

**Table 4-12** Named Characters in *od*

| Value | Name       | Value | Name       | Value | Name             | Value | Name       |
|-------|------------|-------|------------|-------|------------------|-------|------------|
| \000  | <b>nul</b> | \001  | <b>soh</b> | \002  | <b>stx</b>       | \003  | <b>etx</b> |
| \004  | <b>eot</b> | \005  | <b>enq</b> | \006  | <b>ack</b>       | \007  | <b>bel</b> |
| \010  | <b>bs</b>  | \011  | <b>ht</b>  | \012  | <b>lf or nl*</b> | \013  | <b>vt</b>  |
| \014  | <b>ff</b>  | \015  | <b>cr</b>  | \016  | <b>so</b>        | \017  | <b>si</b>  |
| \020  | <b>dle</b> | \021  | <b>dc1</b> | \022  | <b>dc2</b>       | \023  | <b>dc3</b> |
| \024  | <b>dc4</b> | \025  | <b>nak</b> | \026  | <b>syn</b>       | \027  | <b>etb</b> |
| \030  | <b>can</b> | \031  | <b>em</b>  | \032  | <b>sub</b>       | \033  | <b>esc</b> |
| \034  | <b>fs</b>  | \035  | <b>gs</b>  | \036  | <b>rs</b>        | \037  | <b>us</b>  |
| \040  | <b>sp</b>  | \177  | <b>del</b> |       |                  |       |            |

**Note:** The "\012" value may be written either as **lf** or **nl**.

The type specifier character **c** specifies that bytes are interpreted as characters specified by the current setting of the *LC\_CTYPE* locale category. Characters listed in the table in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 5, File Format Notation ("\\", '\a', '\b', '\f', '\n', '\r', '\t', '\v') shall be written as the corresponding escape sequences, except that backslash shall be written as a single backslash and a NUL shall be written as '\0'. Other non-printable characters shall be written as one three-digit octal number for each byte in the character. If the size of a byte on the system is greater than nine bits, the format used for non-printable characters is implementation-dependent. Printable multi-byte characters shall be written in the area corresponding to the first byte of the character; the two-character sequence "\*\*\*" shall be written in the area corresponding to each remaining byte in the character, as an indication that the character is continued. When either the **-j skip** or **-N count** option is specified along with the **c** type specifier, and this results in an attempt to start or finish in the middle of a multi-byte character, the result is implementation-dependent.

The input data shall be manipulated in blocks, where a block is defined as a multiple of the least common multiple of the number of bytes transformed by the specified output types. If the least common multiple is greater than 16, the results are unspecified. Each input block shall be written as transformed by each output type, one per written line, in the order that the output types were specified. If the input block size is larger than the number of bytes transformed by the output type, the output type shall sequentially transform the parts of the input block, and the output from each of the transformations shall be separated by one or more <blank> characters.

If, as a result of the specification of the **-N** option or end-of-file being reached on the last input file, input data only partially satisfies an output type, the input shall be extended sufficiently

27379 with null bytes to write the last byte of the input.

27380 Unless **-A n** is specified, the first output line produced for each input block shall be preceded by  
 27381 the input offset, cumulative across input files, of the next byte to be written. The format of the  
 27382 input offset is unspecified; however, it shall not contain any <blank> characters, shall start at the  
 27383 first character of the output line, and shall be followed by one or more <blank> characters. In  
 27384 addition, the offset of the byte following the last byte written shall be written after all the input  
 27385 data has been processed, but shall not be followed by any <blank> characters.

27386 If no **-A** option is specified, the input offset base is unspecified.

27387 **EXIT STATUS**

27388 The following exit values shall be returned:

27389 0 All input files were processed successfully.

27390 >0 An error occurred.

27391 **CONSEQUENCES OF ERRORS**

27392 Default.

27393 **APPLICATION USAGE**

27394 Applications are warned not to use file names starting with '+' or a first operand starting with  
 27395 a numeric character so that the old functionality can be maintained by implementations, unless  
 27396 they specify one of the new options specified by the ISO/IEC 9945-2:1993 standard. To  
 27397 guarantee that one of these file names is always interpreted as a file name, an application could  
 27398 always specify the address base format with the **-A** option.

27399 **EXAMPLES**

27400 If a file containing 128 bytes with decimal values zero to 127, in increasing order, is supplied as  
 27401 standard input to the command:

27402 `od -A d -t a`

27403 on an implementation using an input block size of 16 bytes, the standard output, independent of  
 27404 the current locale setting, would be similar to:

|       |          |     |     |     |     |     |     |     |     |     |    |     |     |    |    |    |     |
|-------|----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|----|-----|-----|----|----|----|-----|
| 27405 | 00000000 | nul | soh | stx | etx | eot | enq | ack | bel | bs  | ht | nl  | vt  | ff | cr | so | si  |
| 27406 | 0000016  | dle | dc1 | dc2 | dc3 | dc4 | nak | syn | etb | can | em | sub | esc | fs | gs | rs | us  |
| 27407 | 0000032  | sp  | !   | "   | #   | \$  | %   | &   | '   | (   | )  | *   | +   | ,  | -  | .  | /   |
| 27408 | 0000048  | 0   | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9  | :   | ;   | <  | =  | >  | ?   |
| 27409 | 0000064  | @   | A   | B   | C   | D   | E   | F   | G   | H   | I  | J   | K   | L  | M  | N  | O   |
| 27410 | 0000080  | P   | Q   | R   | S   | T   | U   | V   | W   | X   | Y  | Z   | [   | \  | ]  | ^  | _   |
| 27411 | 0000096  | `   | a   | b   | c   | d   | e   | f   | g   | h   | i  | j   | k   | l  | m  | n  | o   |
| 27412 | 0000112  | p   | q   | r   | s   | t   | u   | v   | w   | x   | y  | z   | {   |    | }  | ~  | del |
| 27413 | 0000128  |     |     |     |     |     |     |     |     |     |    |     |     |    |    |    |     |

27414 Note that this volume of IEEE Std. 1003.1-200x allows **nl** or **lf** to be used as the name for the  
 27415 ISO/IEC 646:1991 standard IRV character with decimal value 10. The IRV names this character  
 27416 **lf** (line feed), but traditional implementations have referred to this character as newline (**nl**) and  
 27417 the POSIX locale character set symbolic name for the corresponding character is a <newline>  
 27418 character.

27419 The command:

27420 `od -A o -t o2x2x -n 18`

27421 on a system with 32-bit words and an implementation using an input block size of 16 bytes  
 27422 could write 18 bytes in approximately the following format:

```

27423 00000000 032056 031440 041123 042040 052516 044530 020043 031464
27424 342e 3320 4253 4420 554e 4958 2023 3334
27425 342e3320 42534420 554e4958 20233334
27426 00000020 032472
27427 353a
27428 353a0000
27429 00000022

```

27430 The command:

```
27431 od -A d -t f -t o4 -t x4 -n 24 -j 0x15
```

27432 on a system with 64-bit doubles (for example, the IEEE Std. 754:1985 standard double precision  
 27433 floating-point format) would skip 21 bytes of input data and then write 24 bytes in  
 27434 approximately the following format:

```

27435 00000000 1.0000000000000000e+00 1.5735000000000000e+01
27436 07774000000 000000000000 10013674121 35341217270
27437 3ff00000 00000000 402f3851 eb851eb8
27438 00000016 1.4066823000000000e+02
27439 10030312542 04370303230
27440 40619562 23e18698
27441 00000024

```

#### 27442 RATIONALE

27443 The *od* utility went through several names in early proposals, including *hd*, *xd*, and most recently  
 27444 *hexdump*. There were several objections to all of these based on the following reasons:

- 27445 • The *hd* and *xd* names conflicted with historical utilities that behaved differently.
- 27446 • The *hexdump* description was much more complex than needed for a simple dump utility.
- 27447 • The *od* utility has been available on all historical implementations and there was no need to  
 27448 create a new name for a utility so similar to the historical *od* utility.

27449 The original reasons for not standardizing historical *od* were also fairly widespread. Those  
 27450 reasons are given below along with rationale explaining why the standard developers believe  
 27451 that this version does not suffer from the indicated problem:

- 27452 • The BSD and System V versions of *od* have diverged, and the intersection of features  
 27453 provided by both does not meet the needs of the user community. In fact, the System V  
 27454 version only provides a mechanism for dumping octal bytes and **shorts**, signed and unsigned  
 27455 decimal **shorts**, hexadecimal **shorts**, and ASCII characters. BSD added the ability to dump  
 27456 **floats**, **doubles**, named ASCII characters, and octal, signed decimal, unsigned decimal, and  
 27457 hexadecimal **longs**. The version presented here provides more normalized forms for  
 27458 dumping bytes, **shorts**, **ints**, and **longs** in octal, signed decimal, unsigned decimal, and  
 27459 hexadecimal; **float**, **double**, and **long double**; and named ASCII as well as current locale  
 27460 characters.
- 27461 • It would not be possible to come up with a compatible superset of the BSD and System V  
 27462 flags that met the requirements of the standard developers. The historical default *od* output is  
 27463 the specified default output of this utility. None of the option letters chosen for this version  
 27464 of *od* conflict with any of the options to historical versions of *od*.
- 27465 • On systems with different sizes for **short**, **int**, and **long**, there was no way to ask for dumps  
 27466 of **ints**, even in the BSD version. Because of the way options are named, the namespace could  
 27467 not be extended to solve these problems. This is why the **-t** option was added (with type  
 27468 specifiers more closely matched to the *printf()* formats used in the rest of this volume of

IEEE Std. 1003.1-200x) and the optional field sizes were added to the **d**, **f**, **o**, **u**, and **x** type specifiers. It is also one of the reasons why the historical practice was not mandated as a required obsolescent form of *od*. (Although the old versions of *od* are not listed as an obsolescent form, implementations are urged to continue to recognize the older forms for several more years.) The **a**, **c**, **f**, **o**, and **x** types match the meaning of the corresponding format characters in the historical implementations of *od* except for the default sizes of the fields converted. The **d** format is signed in this volume of IEEE Std. 1003.1-200x to match the *printf()* notation. (Historical versions of *od* used **d** as a synonym for **u** in this version. The System V implementation uses **s** for signed decimal; BSD uses **i** for signed decimal and **s** for null-terminated strings.) Other than **d** and **u**, all of the type specifiers match format characters in the historical BSD version of *od*.

The sizes of the C-language types **char**, **short**, **int**, **long**, **float**, **double**, and **long double** are used even though it is recognized that there may be zero or more than one compiler for the C language on an implementation and that they may use different sizes for some of these types. (For example, one compiler might use 2 bytes **shorts**, 2 bytes **ints**, and 4 bytes **longs**, while another compiler (or an option to the same compiler) uses 2 bytes **shorts**, 4 bytes **ints**, and 4 bytes **longs**.) Nonetheless, there has to be a basic size known by the implementation for these types, corresponding to the values reported by invocations of the *getconf* utility when called with *system\_var* operands {UCHAR\_MAX}, {USHORT\_MAX}, {UINT\_MAX}, and {ULONG\_MAX} for the types **char**, **short**, **int**, and **long**, respectively. There are similar constants required by the ISO C standard, but not required by the System Interfaces volume of IEEE Std. 1003.1-200x or this volume of IEEE Std. 1003.1-200x. They are {FLT\_MANT\_DIG}, {DBL\_MANT\_DIG}, and {LDBL\_MANT\_DIG} for the types **float**, **double**, and **long double**, respectively. If the optional *c89* utility is provided by the implementation and used as specified by this volume of IEEE Std. 1003.1-200x, these are the sizes that would be provided. If an option is used that specifies different sizes for these types, there is no guarantee that the *od* utility is able to interpret binary data output by such a program correctly.

This volume of IEEE Std. 1003.1-200x requires that the numeric values of these lengths be recognized by the *od* utility and that symbolic forms also be recognized. Thus, a portable application can always look at an array of **unsigned long** data elements using *od -t uL*.

- The method of specifying the format for the address field based on specifying a starting offset in a file unnecessarily tied the two together. The **-A** option now specifies the address base and the **-S** option specifies a starting offset.
- It would be difficult to break the dependence on U.S. ASCII to achieve an internationalized utility. It does not seem to be any harder for *od* to dump characters in the current locale than it is for the *ed* or *sed* **l** commands. The **c** type specifier does this without difficulty and is completely compatible with the historical implementations of the **c** format character when the current locale uses a superset of the ISO/IEC 646:1991 standard as a codeset. The **a** type specifier (from the BSD **a** format character) was left as a portable means to dump ASCII (or more correctly ISO/IEC 646:1991 standard (IRV)) so that headers produced by *pax* could be deciphered even on systems that do not use the ISO/IEC 646:1991 standard as a subset of their base codeset.

The use of **"\*\*"** as an indication of continuation of a multi-byte character in **c** specifier output was chosen based on seeing an implementation that uses this method. The continuation bytes have to be marked in a way that is not ambiguous with another single-byte or multi-byte character.

An early proposal used **-S** and **-n**, respectively, for the **-j** and **-N** options eventually selected. These were changed to avoid conflicts with historical implementations.

27518 The original standard specified **-t o2** as the default when no output type was given. This was  
27519 changed to **-t oS** (the length of a **short**) to accommodate a supercomputer implementation that  
27520 historically used 64 bits as its default (and that defined shorts as 64 bits). This change should not  
27521 affect portable applications. The requirement to support lengths of 1, 2, and 4 was added at the  
27522 same time to address an historical implementation that had no two-byte data types in its C  
27523 compiler.

27524 **FUTURE DIRECTIONS**

27525 All option and operand interfaces marked as extensions may be withdrawn in a future issue.

27526 **SEE ALSO**

27527 *sed*

27528 **CHANGE HISTORY**

27529 First released in Issue 2.

27530 **Issue 4**

27531 Aligned with the ISO/IEC 9945-2:1993 standard.

27532 **Issue 4, Version 2**

27533 The description of the **-c** option is made dependent on the current setting of the *LC\_CTYPE*  
27534 category, and a reference to the POSIX locale is deleted.

27535 **Issue 5**

27536 In the description of the **-c** option, the phrase “This is equivalent to **-t c**.” is deleted.

27537 The FUTURE DIRECTIONS section has been modified.

27538 **Issue 6**

27539 The *od* utility is changed to remove the assumption that **short** was a two-byte entity, as per the  
27540 revisions in the IEEE P1003.2b draft standard.

27541 The normative text is reworded to avoid use of the term “must” for application requirements.

27542 **NAME**

27543       paste — merge corresponding or subsequent lines of files

27544 **SYNOPSIS**

27545       paste [-s][-d *list*] *file...*

27546 **DESCRIPTION**

27547       The *paste* utility shall concatenate the corresponding lines of the given input files, and writes the  
27548       resulting lines to standard output.

27549       The default operation of *paste* shall concatenate the corresponding lines of the input files. The  
27550       <newline> character of every line except the line from the last input file shall be replaced with a  
27551       <tab> character.

27552       If an end-of-file condition is detected on one or more input files, but not all input files, *paste* shall  
27553       behave as though empty lines were read from the files on which end-of-file was detected, unless  
27554       the *-s* option is specified.

27555 **OPTIONS**

27556       The *paste* utility shall conform to the System Interface Definitions volume of  
27557       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

27558       The following options shall be supported:

27559       *-d list*       Unless a backslash character appears in *list*, each character in *list* is an element  
27560       specifying a delimiter character. If a backslash character appears in *list*, the  
27561       backslash character and one or more characters following it are an element  
27562       specifying a delimiter character as described below. These elements specify one or  
27563       more delimiters to use, instead of the default <tab> character, to replace the  
27564       <newline> character of the input lines. The elements in *list* shall be used circularly;  
27565       that is, when the list is exhausted the first element from the list is reused. When the  
27566       *-s* option is specified:

- 27567       • The last <newline> character in a file shall not be modified.
- 27568       • The delimiter shall be reset to the first element of list after each *file* operand is  
27569       processed.

27570       When the *-s* option is not specified:

- 27571       • The <newline> characters in the file specified by the last *file* operand shall not  
27572       be modified.
- 27573       • The delimiter shall be reset to the first element of list each time a line is  
27574       processed from each file.

27575       If a backslash character appears in *list*, it and the character following it shall be  
27576       used to represent the following delimiter characters:

27577       \n   <newline> character.

27578       \t   <tab> character.

27579       \\   Backslash character.

27580       \0   Empty string (not a null character). If '\0' is immediately followed by the  
27581       character 'x', the character 'X', or any character defined by the *LC\_CTYPE*  
27582       *digit* keyword (see the System Interface Definitions volume of  
27583       IEEE Std. 1003.1-200x, Chapter 7, Locale), the results are unspecified.

- 27584 If any other characters follow the backslash, the results are unspecified.
- 27585 **-s** Concatenate all of the lines of each separate input file in command line order. The  
27586 <newline> character of every line except the last line in each input file shall be  
27587 replaced with the <tab> character, unless otherwise specified by the **-d** option.
- 27588 **OPERANDS**
- 27589 The following operand shall be supported:
- 27590 **file** A path name of an input file. If **'-'** is specified for one or more of the *files*, the  
27591 standard input shall be used; the standard input shall be read one line at a time,  
27592 circularly, for each instance of **'-'**. Implementations shall support pasting of at  
27593 least 12 *file* operands.
- 27594 **STDIN**
- 27595 The standard input shall be used only if one or more *file* operands is **'-'**. See the INPUT FILES  
27596 section.
- 27597 **INPUT FILES**
- 27598 The input files shall be text files, except that line lengths shall be unlimited.
- 27599 **ENVIRONMENT VARIABLES**
- 27600 The following environment variables shall affect the execution of *paste*:
- 27601 **LANG** Provide a default value for the internationalization variables that are unset or null.  
27602 If *LANG* is unset or null, the corresponding value from the implementation-  
27603 dependent default locale shall be used. If any of the internationalization variables  
27604 contains an invalid setting, the utility shall behave as if none of the variables had  
27605 been defined.
- 27606 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
27607 internationalization variables.
- 27608 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
27609 characters (for example, single-byte as opposed to multi-byte characters in  
27610 arguments and input files).
- 27611 **LC\_MESSAGES**
- 27612 Determine the locale that should be used to affect the format and contents of  
27613 diagnostic messages written to standard error.
- 27614 **XSI NLS\_PATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.
- 27615 **ASYNCHRONOUS EVENTS**
- 27616 Default.
- 27617 **STDOUT**
- 27618 Concatenated lines of input files shall be separated by the <tab> character (or other characters  
27619 under the control of the **-d** option) and terminated by a <newline> character.
- 27620 **STDERR**
- 27621 Used only for diagnostic messages.
- 27622 **OUTPUT FILES**
- 27623 None.
- 27624 **EXTENDED DESCRIPTION**
- 27625 None.

27626 **EXIT STATUS**

27627       The following exit values shall be returned:

27628       0   Successful completion.

27629       >0   An error occurred.

27630 **CONSEQUENCES OF ERRORS**

27631       If one or more input files cannot be opened when the **-s** option is not specified, a diagnostic message shall be written to standard error, but no output is written to standard output. If the **-s** option is specified, the *paste* utility shall provide the default behavior described in Section 1.11 on page 25.

27635 **APPLICATION USAGE**

27636       When the escape sequences of the *list* option-argument are used in a shell script, they must be quoted; otherwise, the shell treats the `'\'` as a special character.

27638       Portable applications should only use the specific backslash escaped delimiters presented in this volume of IEEE Std. 1003.1-200x. Historical implementations treat `'\x'`, where `'x'` is not in this list, as `'x'`, but future implementations are free to expand this list to recognize other common escapes similar to those accepted by *printf* and other standard utilities.

27642       Most of the standard utilities work on text files. The *cut* utility can be used to turn files with arbitrary line lengths into a set of text files containing the same data. The *paste* utility can be used to create (or recreate) files with arbitrary line lengths. For example, if **file** contains long lines:

27645       cut -b 1-500 -n file > file1

27646       cut -b 501- -n file > file2

27647       creates **file1** (a text file) with lines no longer than 500 bytes (plus the <newline> character) and **file2** that contains the remainder of the data from **file**. Note that **file2** is not a text file if there are lines in **file** that are longer than 500 + {LINE\_MAX} bytes. The original file can be recreated from **file1** and **file2** using the command:

27651       paste -d "\0" file1 file2 > file

27652       The commands:

27653       paste -d "\0" ...

27654       paste -d " " ...

27655       are not necessarily equivalent; the latter is not specified by this volume of IEEE Std. 1003.1-200x and may result in an error. The construct `'\0'` is used to mean “no separator” because historical versions of *paste* did not follow the syntax guidelines, and the command:

27658       paste -d " " ...

27659       could not be handled properly by *getopt()*.

27660 **EXAMPLES**

27661       1. Write out a directory in four columns:

27662       ls | paste - - - -

27663       2. Combine pairs of lines from a file into single lines:

27664       paste -s -d "\t\n" file



27665 **RATIONALE**

27666 None.

27667 **FUTURE DIRECTIONS**

27668 None.

27669 **SEE ALSO**27670 *cut, grep, pr*27671 **CHANGE HISTORY**

27672 First released in Issue 2.

27673 **Issue 4**

27674 Aligned with the ISO/IEC 9945-2:1993 standard.

27675 **Issue 6**

27676 The normative text is reworded to avoid use of the term “must” for application requirements.

## 27677 NAME

27678 patch — apply changes to files

## 27679 SYNOPSIS

```
27680 UP patch [-blNR][-c | -e | -n][-d dir][-D define][-i patchfile]
27681 [-o outfile][-p num][-r rejectfile][file]
27682
```

## 27683 DESCRIPTION

27684 The *patch* utility shall read a source (patch) file containing any of the three forms of difference  
 27685 (diff) listings produced by the *diff* utility (normal, context or in the style of *ed*) and apply those  
 27686 differences to a file. By default, *patch* shall read from the standard input.

27687 The *patch* utility shall attempt to determine the type of the *diff* listing, unless overruled by a *-c*,  
 27688 *-e*, or *-n* option.

27689 If the patch file contains more than one patch, *patch* shall attempt to apply each of them as if they  
 27690 came from separate patch files. (In this case, the application shall ensure that the name of the  
 27691 patch file is determinable for each *diff* listing.)

## 27692 OPTIONS

27693 The *patch* utility shall conform to the System Interface Definitions volume of  
 27694 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

27695 The following options shall be supported:

27696 **-b** Save a copy of the original contents of each modified file, before the differences are  
 27697 applied, in a file of the same name with the suffix **.orig** appended to it. If the file  
 27698 already exists, it shall be overwritten; if multiple patches are applied to the same  
 27699 file, the **.orig** file shall be written only for the first patch. When the *-o outfile* option  
 27700 is also specified, *file.orig* shall not be created but, if *outfile* already exists,  
 27701 *outfile.orig* shall be created.

27702 **-c** Interpret the patch file as a context difference (the output of the utility *diff* when  
 27703 the *-c* or *-C* options are specified).

27704 **-d dir** Change the current directory to *dir* before processing as described in the  
 27705 EXTENDED DESCRIPTION section.

27706 **-D define** Mark changes with one of the following C preprocessor constructs:

```
27707 #ifdef define
27708 ...
27709 #endif
27710 #ifndef define
27711 ...
27712 #endif
```

27713 optionally combined with the C preprocessor construct **#else**.

27714 **-e** Interpret the patch file as an *ed* script, rather than a *diff* script.

27715 **-i patchfile** Read the patch information from the file named by the path name *patchfile*, rather  
 27716 than the standard input.

27717 **-l** (The letter ell.) Cause any sequence of <blank> characters in the difference script to  
 27718 match any sequence of <blank> characters in the input file. Other characters shall  
 27719 be matched exactly.

|       |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|-------|----------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 27720 | <b>-n</b>                                                                        | Interpret the script as a normal difference.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 27721 | <b>-N</b>                                                                        | Ignore patches where the differences have already been applied to the file; by default, already-applied patches shall be rejected.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 27722 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27723 | <b>-o outfile</b>                                                                | Instead of modifying the files (specified by the <i>file</i> operand or the difference listings) directly, write a copy of the file referenced by each patch, with the appropriate differences applied, to <i>outfile</i> . Multiple patches for a single file shall be applied to the intermediate versions of the file created by any previous patches, and shall result in multiple, concatenated versions of the file being written to <i>outfile</i> .                                                                                                                                                                                                          |
| 27724 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27725 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27726 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27727 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27728 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27729 | <b>-p num</b>                                                                    | For all path names in the patch file that indicate the names of files to be patched, delete <i>num</i> path name components from the beginning of each path name. If the path name in the patch file is absolute, any leading slashes shall be considered the first component (that is, <b>-p 1</b> shall remove the leading slashes). Specifying <b>-p 0</b> shall cause the full path name to be used. If <b>-p</b> is not specified, only the basename (the final path name component) shall be used.                                                                                                                                                             |
| 27730 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27731 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27732 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27733 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27734 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27735 | <b>-R</b>                                                                        | Reverse the sense of the patch script; that is, assume that the difference script was created from the new version to the old version. The <b>-R</b> option cannot be used with <i>ed</i> scripts. The <i>patch</i> utility attempts to reverse each portion of the script before applying it. Rejected differences shall be saved in swapped format. If this option is not specified, and until a portion of the patch file is successfully applied, <i>patch</i> attempts to apply each portion in its reversed sense as well as in its normal sense. If the attempt is successful, the user shall be prompted to determine if the <b>-R</b> option should be set. |
| 27736 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27737 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27738 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27739 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27740 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27741 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27742 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27743 | <b>-r rejectfile</b>                                                             | Override the default reject file name. In the default case, the reject file shall have the same name as the output file, with the suffix <b>.rej</b> appended to it; see <b>Patch Application</b> on page 729.                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 27744 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27745 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27746 | <b>OPERANDS</b>                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27747 | The following operand shall be supported:                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27748 | <i>file</i>                                                                      | A path name of a file to patch.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27749 | <b>STDIN</b>                                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27750 | See the INPUT FILES section.                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27751 | <b>INPUT FILES</b>                                                               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27752 | Input files shall be text files.                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27753 | <b>ENVIRONMENT VARIABLES</b>                                                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27754 | The following environment variables shall affect the execution of <i>patch</i> : |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27755 | <i>LANG</i>                                                                      | Provide a default value for the internationalization variables that are unset or null. If <i>LANG</i> is unset or null, the corresponding value from the implementation-dependent default locale shall be used. If any of the internationalization variables contains an invalid setting, the utility shall behave as if none of the variables had been defined.                                                                                                                                                                                                                                                                                                     |
| 27756 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27757 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27758 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27759 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27760 | <i>LC_ALL</i>                                                                    | If set to a non-empty string value, override the values of all the other internationalization variables.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 27761 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27762 | <i>LC_CTYPE</i>                                                                  | Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments and input files).                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 27763 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 27764 |                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

27765 **LC\_MESSAGES**  
 27766 Determine the locale that should be used to affect the format and contents of  
 27767 diagnostic messages written to standard error and informative messages written to  
 27768 standard output.

27769 XSI **NLSPATH** Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

27770 **LC\_TIME** Determine the locale for recognizing the format of file timestamps written by the  
 27771 *diff* utility in a context-difference input file.

27772 **ASYNCHRONOUS EVENTS**  
 27773 Default.

27774 **STDOUT**  
 27775 Not used.

27776 **STDERR**  
 27777 Used for diagnostic and informational messages.

27778 **OUTPUT FILES**  
 27779 The output of the *patch* utility, the save files (**.orig** suffixes) and the reject files (**.rej** suffixes) shall  
 27780 be text files.

27781 **EXTENDED DESCRIPTION**  
 27782 A patchfile may contain patching instructions for more than one file; file names shall be  
 27783 determined as specified in **File Name Determination** on page 729. When the **-b** option is  
 27784 specified, for each patched file, the original shall be saved in a file of the same name with the  
 27785 suffix **.orig** appended to it.

27786 For each patched file, a reject file may also be created as noted in **Patch Application** on page 729.  
 27787 In the absence of a **-r** option, the name of this file shall be formed by appending the suffix **.rej** to  
 27788 the original file name.

27789 **Patchfile Format**  
 27790 The patch file shall contain zero or more lines of header information followed by one or more  
 27791 patches. Each patch shall contain zero or more lines of file name identification in the format  
 27792 produced by *diff -c*, and one or more sets of *diff* output, which are customarily called *hunks*.

27793 The *patch* utility shall recognize the following expression in the header information:

27794 **Index: *pathname***  
 27795 The file to be patched is named *pathname*.

27796 If all lines (including headers) within a patch begin with the same leading sequence of <blank>  
 27797 characters, the *patch* utility shall remove this sequence before proceeding. Within each patch, if  
 27798 the type of difference is context, the *patch* utility shall recognize the following expressions:

27799 **\*\*\* *filename* timestamp**  
 27800 The patches arose from *filename*.

27801 **--- *filename* timestamp**  
 27802 The patches should be applied to *filename*.

27803 Each hunk within a patch shall be the *diff* output to change a line range within the original file.  
 27804 The line numbers for successive hunks within a patch shall occur in ascending order.

27805 **File Name Determination**

27806 If no *file* operand is specified, *patch* shall perform the following steps to determine the file name  
 27807 to use:

- 27808 1. If the type of *diff* is context, the *patch* utility shall delete path name components (as  
 27809 specified by the **-p** option) from the file name on the line beginning with "\*\*\*\*", then test  
 27810 for the existence of this file relative to the current directory (or the directory specified with  
 27811 the **-d** option). If the file exists, the *patch* utility shall use this file name.
- 27812 2. If the type of *diff* is context, the *patch* utility shall delete the path name components (as  
 27813 specified by the **-p** option) from the file name on the line beginning with "——", then test  
 27814 for the existence of this file relative to the current directory (or the directory specified with  
 27815 the **-d** option). If the file exists, the *patch* utility shall use this file name.
- 27816 3. If the header information contains a line beginning with the string **Index:**, the *patch* utility  
 27817 shall delete path name components (as specified by the **-p** option) from this line, then test  
 27818 for the existence of this file relative to the current directory (or the directory specified with  
 27819 the **-d** option). If the file exists, the *patch* utility shall use this file name.
- 27820 XSI 4. If an **SCCS** directory exists in the current directory, *patch* shall attempt to perform a **get -e**  
 27821 **SCCS/s.filename** command to retrieve an editable version of the file.
- 27822 5. The *patch* utility shall write a prompt to standard output and request a file name  
 27823 interactively from the controlling terminal (for example, **/dev/tty**).

27824 **Patch Application**

27825 If the **-c**, **-e**, or **-n** option is present, the *patch* utility shall interpret information within each hunk  
 27826 as a context difference, an *ed* difference or a normal difference, respectively. In the absence of  
 27827 any of these options, the *patch* utility shall determine the type of difference based on the format  
 27828 of information within the hunk.

27829 For each hunk, the *patch* utility shall begin to search for the place to apply the patch at the line  
 27830 number at the beginning of the hunk, plus or minus any offset used in applying the previous  
 27831 hunk. If lines matching the hunk context are not found, *patch* shall scan both forwards and  
 27832 backwards at least 1 000 bytes for a set of lines that match the hunk context.

27833 If no such place is found and it is a context difference, then another scan shall take place,  
 27834 ignoring the first and last line of context. If that fails, the first two and last two lines of context  
 27835 shall be ignored and another scan shall be made. Implementations may search more extensively  
 27836 for installation locations.

27837 If no location can be found, the *patch* utility shall append the hunk to the reject file. The rejected  
 27838 hunk shall be written in context-difference format regardless of the format of the patch file. If the  
 27839 input was a normal or *ed*-style difference, the reject file may contain differences with zero lines  
 27840 of context. The line numbers on the hunks in the reject file may be different from the line  
 27841 numbers in the patch file since they shall reflect the approximate locations for the failed hunks in  
 27842 the new file rather than the old one.

27843 If the type of patch is an *ed* diff, the implementation may accomplish the patching by invoking  
 27844 the *ed* utility.

27845 **EXIT STATUS**

27846 The following exit values shall be returned:

- 27847 0 Successful completion.

27848           1   One or more lines were written to a reject file.

27849           >1   An error occurred.

## 27850 CONSEQUENCES OF ERRORS

27851           Patches that cannot be correctly placed in the file shall be written to a reject file.

## 27852 APPLICATION USAGE

27853           The **-R** option does not work with *ed* scripts because there is too little information to reconstruct the reverse operation.

27855           The **-p** option makes it possible to customize a patchfile to local user directory structures without manually editing the patchfile. For example, if the file name in the patch file was:

27857           /curds/whey/src/blurfl/blurfl.c

27858           Setting **-p 0** gives the entire path name unmodified; **-p 1** gives:

27859           curds/whey/src/blurfl/blurfl.c

27860           without the leading slash, **-p 4** gives:

27861           blurfl/blurfl.c

27862           and not specifying **-p** at all gives:

27863           blurfl.c .

27864           When using **-b** in some file system implementations, the saving of a **.orig** file may produce unwanted results. In the case of 12, 13, or 14-character file names, on file systems supporting 14-character maximum file names, the **.orig** file overwrites the new file.

27867           Application writers should note that this utility need not be provided on systems that do not support the User Portability Utilities option.

## 27869 EXAMPLES

27870           None.

## 27871 RATIONALE

27872           Some of the functionality in historical *patch* implementations was not specified. The following documents those features present in historical implementations that have not been specified.

27874           A deleted piece of functionality was the '+' pseudo-option allowing an additional set of options and a patch file operand to be given. This was seen as being insufficiently useful to standardize.

27876           In historical implementations, if the string "Prereq: " appeared in the header, the *patch* utility would search for the corresponding version information (the string specified in the header, delimited by <blank>s or the beginning or end of a line or the file) anywhere in the original file. This was deleted as too simplistic and insufficiently trustworthy a mechanism to standardize. For example, if:

27881           Prereq: 1.2

27882           were in the header, the presence of a delimited 1.2 anywhere in the file would satisfy the prerequisite.

27884           The following options were dropped from historical implementations of *patch* as insufficiently useful to standardize:

27886           **-b**       The **-b** option historically provided a method for changing the name extension of the backup file from the default **.orig**. This option has been modified and retained in this volume of IEEE Std. 1003.1-200x.

|       |                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------|--------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 27889 | <b>-F</b>                | The <b>-F</b> option specified the number of lines of a context diff to ignore when searching for a place to install a patch.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 27890 |                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27891 | <b>-f</b>                | The <b>-f</b> option historically caused <i>patch</i> not to request additional information from the user.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 27892 |                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27893 | <b>-r</b>                | The <b>-r</b> option historically provided a method of overriding the extension of the reject file from the default <b>.rej</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 27894 |                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27895 | <b>-s</b>                | The <b>-s</b> option historically caused <i>patch</i> to work silently unless an error occurred.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 27896 | <b>-x</b>                | The <b>-x</b> option historically set internal debugging flags.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27897 |                          | In some file system implementations, the saving of a <b>.orig</b> file may produce unwanted results. In the case of 12, 13, or 14-character file names (on file systems supporting 14-character maximum file names), the <b>.orig</b> file overwrites the new file. The reject file may also exceed this file name limit. It was suggested, due to some historical practice, that a tilde ('~') suffix be used instead of <b>.orig</b> and some other character instead of the <b>.rej</b> suffix. This was rejected because it is not obvious to the user which file is which. The suffixes <b>.orig</b> and <b>.rej</b> are clearer and more understandable. |
| 27898 |                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27899 |                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27900 |                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27901 |                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27902 |                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27903 |                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27904 |                          | The <b>-b</b> option has the opposite sense in some historical implementations—do not save the <b>.orig</b> file. The default case here is not to save the files, making <i>patch</i> behave more consistently with the other standard utilities.                                                                                                                                                                                                                                                                                                                                                                                                              |
| 27905 |                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27906 |                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27907 |                          | The <b>-w</b> option in early proposals was changed to <b>-l</b> to match historical practice.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 27908 |                          | The <b>-N</b> option was included because without it, a non-interactive application cannot reject previously applied patches. For example, if a user is piping the output of <i>diff</i> into the <i>patch</i> utility, and the user only wants to patch a file to a newer version non-interactively, the <b>-N</b> option is required.                                                                                                                                                                                                                                                                                                                        |
| 27909 |                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27910 |                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27911 |                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27912 |                          | Changes to the <b>-l</b> option description were proposed to allow matching across <newline>s in addition to just <blank>s. Since this is not historical practice, and since some ambiguities could result, it is suggested that future developments in this area utilize another option letter, such as <b>-L</b> .                                                                                                                                                                                                                                                                                                                                           |
| 27913 |                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27914 |                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27915 |                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27916 | <b>FUTURE DIRECTIONS</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27917 |                          | None.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 27918 | <b>SEE ALSO</b>          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27919 |                          | <i>ed</i> , <i>diff</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 27920 | <b>CHANGE HISTORY</b>    |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27921 |                          | First released in Issue 4.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 27922 | <b>Issue 5</b>           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27923 |                          | FUTURE DIRECTIONS section added.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 27924 | <b>Issue 6</b>           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27925 |                          | This utility is now marked as part of the User Portability Utilities option.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 27926 |                          | The description of the <b>-D</b> option and the steps in <b>File Name Determination</b> on page 729 are changed to match historical practice as defined in the IEEE P1003.2b draft standard.                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 27927 |                          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 27928 |                          | The normative text is reworded to avoid use of the term “must” for application requirements.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |

27929 **NAME**

27930 pathchk — check path names

27931 **SYNOPSIS**27932 pathchk [-p] *pathname...*27933 **DESCRIPTION**

27934 The *pathchk* utility shall check that one or more path names are valid (that is, they could be used  
 27935 to access or create a file without causing syntax errors) and portable (that is, no file name  
 27936 truncation results). More extensive portability checks are provided by the **-p** option.

27937 By default, the *pathchk* utility shall check each component of each *pathname* operand based on the  
 27938 underlying file system. A diagnostic shall be written for each *pathname* operand that:

- 27939 • Is longer than {PATH\_MAX} bytes (see **Path Name Variable Values** in the System Interface  
 27940 Definitions volume of IEEE Std. 1003.1-200x, Chapter 13, Headers, <limits.h>)
- 27941 • Contains any component longer than {NAME\_MAX} bytes in its containing directory
- 27942 • Contains any component in a directory that is not searchable
- 27943 • Contains any character in any component that is not valid in its containing directory

27944 The format of the diagnostic message is not specified, but shall indicate the error detected and  
 27945 the corresponding *pathname* operand.

27946 It shall not be considered an error if one or more components of a *pathname* operand do not exist  
 27947 as long as a file matching the path name specified by the missing components could be created  
 27948 that does not violate any of the checks specified above.

27949 **OPTIONS**

27950 The *pathchk* utility shall conform to the System Interface Definitions volume of  
 27951 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

27952 The following option shall be supported:

- 27953 **-p** Instead of performing checks based on the underlying file system, write a  
 27954 diagnostic for each *pathname* operand that:
  - 27955 • Is longer than {\_POSIX\_PATH\_MAX} bytes (see **Minimum Values** in the  
 27956 System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 13,  
 27957 Headers, <limits.h>)
  - 27958 • Contains any component longer than {\_POSIX\_NAME\_MAX} bytes
  - 27959 • Contains any character in any component that is not in the portable file name  
 27960 character set

27961 **OPERANDS**

27962 The following operand shall be supported:

27963 *pathname* A path name to be checked.

27964 **STDIN**

27965 Not used.

27966 **INPUT FILES**

27967 None.



27968 **ENVIRONMENT VARIABLES**

27969 The following environment variables shall affect the execution of *pathchk*:

27970 **LANG** Provide a default value for the internationalization variables that are unset or null.  
 27971 If *LANG* is unset or null, the corresponding value from the implementation-  
 27972 dependent default locale shall be used. If any of the internationalization variables  
 27973 contains an invalid setting, the utility shall behave as if none of the variables had  
 27974 been defined.

27975 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 27976 internationalization variables.

27977 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 27978 characters (for example, single-byte as opposed to multi-byte characters in  
 27979 arguments).

27980 **LC\_MESSAGES**  
 27981 Determine the locale that should be used to affect the format and contents of  
 27982 diagnostic messages written to standard error.

27983 **NSI** **NLS\_PATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

27984 **ASYNCHRONOUS EVENTS**

27985 Default.

27986 **STDOUT**

27987 Not used.

27988 **STDERR**

27989 Used only for diagnostic messages.

27990 **OUTPUT FILES**

27991 None.

27992 **EXTENDED DESCRIPTION**

27993 None.

27994 **EXIT STATUS**

27995 The following exit values shall be returned:

27996 0 All *pathname* operands passed all of the checks.

27997 >0 An error occurred.

27998 **CONSEQUENCES OF ERRORS**

27999 Default.

28000 **APPLICATION USAGE**

28001 The *test* utility can be used to determine whether a given path name names an existing file; it  
 28002 does not, however, give any indication of whether or not any component of the path name was  
 28003 truncated in a directory where the *\_POSIX\_NO\_TRUNC* feature is not in effect. The *pathchk*  
 28004 utility does not check for file existence; it performs checks to determine if a path name does exist  
 28005 or could be created with no path name component truncation.

28006 The *noclobber* option in the shell (see the *set* on page 117 special built-in) can be used to  
 28007 atomically create a file. As with all file creation semantics in the System Interfaces volume of  
 28008 IEEE Std. 1003.1-200x, it guarantees atomic creation, but still depends on applications to agree  
 28009 on conventions and cooperate on the use of files after they have been created.

28010 **EXAMPLES**

28011 To verify that all path names in an imported data interchange archive are legitimate and  
 28012 unambiguous on the current system:

```
28013 pax -f archive | sed -e '/' == './s///' | xargs pathchk
28014 if [$? -eq 0]
28015 then
28016 pax -r -f archive
28017 else
28018 echo Investigate problems before importing files.
28019 exit 1
28020 fi
```

28021 To verify that all files in the current directory hierarchy could be moved to any system  
 28022 conforming to the System Interfaces volume of IEEE Std. 1003.1-200x that also supports the *pax*  
 28023 utility:

```
28024 find . -print | xargs pathchk -p
28025 if [$? -eq 0]
28026 then
28027 pax -w -f archive .
28028 else
28029 echo Portable archive cannot be created.
28030 exit 1
28031 fi
```

28032 To verify that a user-supplied path name names a readable file and that the application can  
 28033 create a file extending the given path without truncation and without overwriting any existing  
 28034 file:

```
28035 case $- in
28036 *C*) reset="";;
28037 *) reset="set +C"
28038 set -C;;
28039 esac
28040 test -r "$path" && pathchk "$path.out" &&
28041 rm "$path.out" > "$path.out"
28042 if [$? -ne 0]; then
28043 printf "%s: %s not found or %s.out fails \
28044 creation checks.\n" $0 "$path" "$path"
28045 $reset # Reset the noclobber option in case a trap
28046 # on EXIT depends on it.
28047 exit 1
28048 fi
28049 $reset
28050 PROCESSING < "$path" > "$path.out"
```

28051 The following assumptions are made in this example:

- 28052 1. **PROCESSING** represents the code that is used by the application to use **\$path** once it is  
 28053 verified that **\$path.out** works as intended.
- 28054 2. The state of the *noclobber* option is unknown when this code is invoked and should be set  
 28055 on exit to the state it was in when this code was invoked. (The **reset** variable is used in this  
 28056 example to restore the initial state.)

## 3. Note the usage of:

```
rm "$path.out" > "$path.out"
```

- a. The *pathchk* command has already verified, at this point, that **\$path.out** is not truncated.
- b. With the *noclobber* option set, the shell verifies that **\$path.out** does not already exist before invoking *rm*.
- c. If the shell succeeded in creating **\$path.out**, *rm* removes it so that the application can create the file again in the **PROCESSING** step.
- d. If the **PROCESSING** step wants the file to exist already when it is invoked, the:

```
rm "$path.out" > "$path.out"
```

should be replaced with:

```
> "$path.out"
```

which verifies that the file did not already exist, but leaves **\$path.out** in place for use by **PROCESSING**.

**RATIONALE**

The *pathchk* utility is new, commissioned for this volume of IEEE Std. 1003.1-200x. It, along with the *set -C(noclobber)* option added to the shell, replaces the *mktemp*, *validfnam*, and *create* utilities that appeared in early proposals. All of these utilities were attempts to solve several common problems:

- Verify the validity (for several different definitions of “valid”) of a path name supplied by a user, generated by an application, or imported from an external source.
- Atomically create a file.
- Perform various string handling functions to generate a temporary file name.

The *create* utility, included in an early proposal, provided checking and atomic creation in a single invocation of the utility; these are orthogonal issues and need not be grouped into a single utility. Note that the *noclobber* option also provides a way of creating a lock for process synchronization; since it provides an atomic *create*, there is no race between a test for existence and the following creation if it did not exist.

Having a function like *tmpnam()* in the ISO C standard is important in many high-level languages. The shell programming language, however, has built-in string manipulation facilities, making it very easy to construct temporary file names. The names needed obviously depend on the application, but are frequently of a form similar to:

```
$TMPDIR/application_abbreviation$$$.suffix
```

In cases where there is likely to be contention for a given suffix, a simple shell *for* or *while* loop can be used with the shell *noclobber* option to create a file without risk of collisions, as long as applications trying to use the same file name namespace are cooperating on the use of files after they have been created.

**FUTURE DIRECTIONS**

None.

28096 **SEE ALSO**

28097           *test*, Section 2.7 on page 60

28098 **CHANGE HISTORY**

28099           First released in Issue 4.

## 28100 NAME

28101 pax — portable archive interchange

## 28102 SYNOPSIS

28103 pax [-cdnv][-H|-L][-f archive][-s replstr]...[pattern...]

28104 pax -r[-cdiknuv][-H|-L][-f archive][-o options]...[-p string]...  
28105 [-s replstr]...[pattern...]28106 pax -w[-dituvX][-H|-L][-b blocksize][[-a][-f archive][-o options]...  
28107 [-s replstr]...[-x format][file...]28108 pax -r -w[-diklntuvX][-H|-L][-p string]...[-s replstr]...  
28109 [file...] directory

## 28110 DESCRIPTION

28111 *Notes to Reviewers*28112 *This section with side shading will not appear in the final copy. - Ed.*28113 *pax* has been extensively changed due to the merger with .2b.28114 The *pax* utility shall read, write, and write lists of the members of archive files and copy  
28115 directory hierarchies. A variety of archive formats shall be supported; see the *-x format* option.28116 The action to be taken depends on the presence of the *-r* and *-w* options. The four combinations  
28117 of *-r* and *-w* are referred to as the four modes of operation: **list**, **read**, **write**, and **copy** modes,  
28118 corresponding respectively to the four forms shown in the SYNOPSIS section.28119 **list** In **list** mode (when neither *-r* nor *-w* are specified), *pax* shall write the names of  
28120 the members of the archive file read from the standard input, with path names  
28121 matching the specified patterns, to standard output. If a named file is of type  
28122 directory, the file hierarchy rooted at that file shall be listed as well.28123 **read** In **read** mode (when *-r* is specified, but *-w* is not), *pax* shall extract the members of  
28124 the archive file read from the standard input, with path names matching the  
28125 specified patterns. If an extracted file is of type directory, the file hierarchy rooted  
28126 at that file shall be extracted as well. The extracted files shall be created relative to  
28127 the current file hierarchy.28128 The ownership, access, and modification times, and file mode of the restored files  
28129 are discussed under the *-p* option.28130 **write** In **write** mode (when *-w* is specified, but *-r* is not), *pax* shall write the contents of  
28131 the *file* operands to the standard output in an archive format. If no *file* operands are  
28132 specified, a list of files to copy, one per line, shall be read from the standard input.  
28133 A file of type directory shall include all of the files in the file hierarchy rooted at the  
28134 file.28135 **copy** In **copy** mode (when both *-r* and *-w* are specified), *pax* shall copy the *file* operands  
28136 to the destination directory.28137 If no *file* operands are specified, a list of files to copy, one per line, shall be read  
28138 from the standard input. A file of type directory shall include all of the files in the  
28139 file hierarchy rooted at the file.28140 The effect of the **copy** shall be as if the copied files were written to an archive file  
28141 and then subsequently extracted, except that there may be hard links between the  
28142 original and the copied files. If the destination directory is a subdirectory of one of  
28143 the files to be copied, the results are unspecified. If the destination directory is a

28144 file of a type not defined by the System Interfaces volume of IEEE Std. 1003.1-200x,  
 28145 the results are implementation-dependent; otherwise, it shall be an error for the file  
 28146 named by the *directory* operand not to exist, not be writable by the user, or not be a  
 28147 file of type *directory*.

28148 In **read** or **copy** modes, if intermediate directories are necessary to extract an archive member,  
 28149 *pax* shall perform actions equivalent to the *mkdir()* function defined in the System Interfaces  
 28150 volume of IEEE Std. 1003.1-200x, called with the following arguments:

- 28151 • The intermediate directory used as the *path* argument
- 28152 • The value of the bitwise-inclusive OR of S\_IRWXU, S\_IRWXG, and S\_IRWXO as the *mode*  
 28153 argument

28154 If any specified *pattern* or *file* operands are not matched by at least one file or archive member,  
 28155 *pax* shall write a diagnostic message to standard error for each one that did not match and exit  
 28156 with a non-zero exit status.

28157 The archive formats described in the EXTENDED DESCRIPTION section shall be automatically  
 28158 detected on input. The default output archive format shall be implementation-dependent.

28159 A single archive can span multiple files. The *pax* utility shall determine, in an implementation-  
 28160 dependent manner, what file to read or write as the next file.

28161 If the selected archive format supports the specification of linked files, it shall be an error if these  
 28162 files cannot be linked when the archive is extracted. For archive formats that do not store file  
 28163 contents with each name that causes a hard link, if the file that contains the data is not extracted  
 28164 during this *pax* session, either the data shall be restored from the original file, or a diagnostic  
 28165 message shall be displayed with the name of a file that can be used to extract the data. In  
 28166 traversing directories, *pax* shall detect infinite loops; that is, entering a previously visited  
 28167 directory that is an ancestor of the last file visited. When it detects an infinite loop, *pax* shall  
 28168 write a diagnostic message to standard error and shall terminate.

## 28169 OPTIONS

28170 The *pax* utility shall conform to the System Interface Definitions volume of  
 28171 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines, except that the order of  
 28172 presentation of the **-o**, **-p**, and **-s** options is significant.

28173 The following options shall be supported:

- 28174 **-r** Read an archive file from standard input.
- 28175 **-w** Write files to the standard output in the specified archive format.
- 28176 **-a** Append files to the end of the archive. It is implementation-dependent which  
 28177 devices on the system support appending. Additional file formats unspecified by  
 28178 this volume of IEEE Std. 1003.1-200x may impose restrictions on appending.
- 28179 **-b *blocksize*** Block the output at a positive decimal integer number of bytes per write to the  
 28180 archive file. Devices and archive formats may impose restrictions on blocking.  
 28181 Blocking shall be automatically determined on input. Portable applications shall  
 28182 not specify a *blocksize* value larger than 32 256. Default blocking when creating  
 28183 archives depends on the archive format. (See the **-x** option below.)
- 28184 **-c** Match all file or archive members except those specified by the *pattern* or *file*  
 28185 operands.
- 28186 **-d** Cause files of type *directory* being copied or archived or archive members of type  
 28187 *directory* being extracted or listed to match only the file or archive member itself  
 28188 and not the file hierarchy rooted at the file.

|       |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|-------|-------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 28189 | <b>-f archive</b> | Specify the path name of the input or output archive, overriding the default standard input (in <b>list</b> or <b>read</b> modes) or standard output ( <b>write</b> mode).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 28190 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28191 | <b>-H</b>         | If a symbolic link referencing a file of type directory is specified on the command line, <i>pax</i> shall archive the file hierarchy rooted in the file referenced by the link, using the name of the link as the root of the file hierarchy. The default behavior shall be to archive the symbolic link itself.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 28192 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28193 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28194 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28195 | <b>-i</b>         | Interactively rename files or archive members. For each archive member matching a <i>pattern</i> operand or file matching a <i>file</i> operand, a prompt shall be written to the file <b>/dev/tty</b> . The prompt shall contain the name of the file or archive member, but the format is otherwise unspecified. A line shall then be read from <b>/dev/tty</b> . If this line is blank, the file or archive member shall be skipped. If this line consists of a single period, the file or archive member shall be processed with no modification to its name. Otherwise, its name shall be replaced with the contents of the line. The <i>pax</i> utility shall immediately exit with a non-zero exit status if end-of-file is encountered when reading a response or if <b>/dev/tty</b> cannot be opened for reading and writing. |
| 28196 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28197 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28198 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28199 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28200 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28201 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28202 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28203 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28204 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28205 |                   | The results of extracting a hard link to a file that has been renamed during extraction are unspecified.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 28206 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28207 | <b>-k</b>         | Prevent the overwriting of existing files.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 28208 | <b>-l</b>         | (The letter ell.) In <b>copy</b> mode, hard links shall be made between the source and destination file hierarchies whenever possible.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 28209 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28210 | <b>-L</b>         | If a symbolic link referencing a file of type directory is specified on the command line or encountered during the traversal of a file hierarchy, <i>pax</i> shall archive the file hierarchy rooted in the file referenced by the link, using the name of the link as the root of the file hierarchy. The default behavior shall be to archive the symbolic link itself.                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 28211 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28212 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28213 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28214 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28215 | <b>-n</b>         | Select the first archive member that matches each <i>pattern</i> operand. No more than one archive member shall be matched for each pattern (although members of type directory shall still match the file hierarchy rooted at that file).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 28216 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28217 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28218 | <b>-o options</b> | Provide information to the implementation to modify the algorithm for extracting or writing files. The value of <i>options</i> shall consist of one or more comma-separated keywords of the form:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 28219 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28220 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28221 |                   | <i>keyword</i> [[:]= <i>value</i> ][, <i>keyword</i> [[:]= <i>value</i> ], ...]                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28222 |                   | Some keywords apply only to certain file formats, as indicated with each description. Use of keywords that are inapplicable to the file format being processed produces undefined results.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 28223 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28224 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28225 |                   | Keywords in the <i>options</i> argument shall be a string that would be a valid portable file name as described in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 3.282, Portable File Name Character Set.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 28226 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28227 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28228 | <b>Note:</b>      | Keywords are not expected to be file names, merely to follow the same character composition rules as portable file names.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 28229 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28230 |                   | Keywords can be preceded with white space. The <i>value</i> field shall consist of zero or more characters; within <i>value</i> , the application shall precede any literal comma with a backslash, which shall be ignored, but preserves the comma as part of <i>value</i> . A comma as the final character, or a comma followed solely by white space as the final characters, in <i>options</i> shall be ignored. Multiple <b>-o</b> options can be specified; if                                                                                                                                                                                                                                                                                                                                                                   |
| 28231 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28232 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28233 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28234 |                   |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |

keywords given to these multiple **-o** options conflict, the keywords and values appearing later in command line sequence shall take precedence and the earlier shall be silently ignored. The following keyword values of *options* shall be supported for the file formats as indicated:

**delete=pattern**

(Applicable only to the **-x** *pax* format.) When used in **write** or **copy** mode, *pax* shall omit from extended header records that it produces any keywords matching the string pattern. When used in **read** or **list** mode, *pax* shall ignore any keywords matching the string pattern in the extended header records. In both cases, matching shall be performed using the pattern matching notation described in Section 2.13.1 on page 92 and Section 2.13.2 on page 93. For example:

**-o delete=security.\***

would suppress security-related information. See **pax Extended Header** on page 750 for extended header record keyword usage.

**exthdr.name=string**

(Applicable only to the **-x** *pax* format.) This keyword allows user control over the name that is written into the **ustar** header blocks for the extended header produced under the circumstances described in **pax Header Block** on page 749. The name shall be the contents of *string*, after the following character substitutions have been made:

| <i>string</i><br>Includes: | Replaced By:                                                                                                        |
|----------------------------|---------------------------------------------------------------------------------------------------------------------|
| %d                         | The directory name of the file, equivalent to the result of the <i>dirname</i> utility on the translated path name. |
| %f                         | The file name of the file, equivalent to the result of the <i>basename</i> utility on the translated path name.     |
| %%                         | A ' % ' character.                                                                                                  |

Any other ' % ' characters in *string* produce undefined results.

If no **-o exthdr.name=string** is specified, *pax* shall use the following default value:

%d/PaxHeaders/%f

**globexthdr.name=string**

(Applicable only to the **-x** *pax* format.) When used in **write** or **copy** mode with the appropriate options, *pax* creates global extended header records with **ustar** header blocks that will be treated as regular files by previous versions of *pax*. This keyword allows user control over the name that is written into the **ustar** header blocks for global extended header records. The name shall be the contents of *string*, after the following character substitutions have been made:

| <i>string</i><br>Includes: | Replaced By:                                                                                                       |
|----------------------------|--------------------------------------------------------------------------------------------------------------------|
| %n                         | An integer that represents the sequence number of the global extended header record in the archive, starting at 1. |
| %%                         | A ' % ' character.                                                                                                 |



|       |                                                                                                       |                                                                                             |
|-------|-------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------|
| 28279 | Any other ' % ' characters in <i>string</i> produce undefined results.                                |                                                                                             |
| 28280 | If no <b>-o globexthdr.name=string</b> is specified, <i>pax</i> shall use the following               |                                                                                             |
| 28281 | default value:                                                                                        |                                                                                             |
| 28282 | \$TMPDIR/GlobalHead.%n                                                                                |                                                                                             |
| 28283 | where \$TMPDIR represents the value of the <i>TMPDIR</i> environment variable. If                     |                                                                                             |
| 28284 | <i>TMPDIR</i> is not set, <i>pax</i> shall use <b>/tmp</b> .                                          |                                                                                             |
| 28285 | <b>invalid=action</b>                                                                                 |                                                                                             |
| 28286 | (Applicable only to the <b>-x pax</b> format.) This keyword allows user control over                  |                                                                                             |
| 28287 | the action <i>pax</i> takes upon encountering values in an extended header record                     |                                                                                             |
| 28288 | that, in <b>read</b> or <b>copy</b> mode, are invalid in the destination hierarchy or, in <b>list</b> |                                                                                             |
| 28289 | mode, cannot be written in the codeset and current locale of the                                      |                                                                                             |
| 28290 | implementation. The following are invalid values that shall be recognized by                          |                                                                                             |
| 28291 | <i>pax</i> :                                                                                          |                                                                                             |
| 28292 | — In <b>read</b> or <b>copy</b> mode, a file name or link name that contains character                |                                                                                             |
| 28293 | encodings invalid in the destination hierarchy. (For example, the name                                |                                                                                             |
| 28294 | may contain embedded NULs.)                                                                           |                                                                                             |
| 28295 | — In <b>read</b> or <b>copy</b> mode, a file name or link name that is longer than the                |                                                                                             |
| 28296 | maximum allowed in the destination hierarchy (for either a path name                                  |                                                                                             |
| 28297 | component or the entire path name).                                                                   |                                                                                             |
| 28298 | — In <b>list</b> mode, any character string value (file name, link name, user name,                   |                                                                                             |
| 28299 | and so on) that cannot be written in the codeset and current locale of the                            |                                                                                             |
| 28300 | implementation.                                                                                       |                                                                                             |
| 28301 | The following mutually-exclusive values of the <i>action</i> argument are                             |                                                                                             |
| 28302 | supported:                                                                                            |                                                                                             |
| 28303 | <b>bypass</b>                                                                                         | In <b>read</b> or <b>copy</b> mode, <i>pax</i> shall bypass the file, causing no            |
| 28304 |                                                                                                       | change to the destination hierarchy. In <b>list</b> mode, <i>pax</i> shall write            |
| 28305 |                                                                                                       | all requested valid values for the file, but its method for writing                         |
| 28306 |                                                                                                       | invalid values is unspecified.                                                              |
| 28307 | <b>rename</b>                                                                                         | In <b>read</b> or <b>copy</b> mode, <i>pax</i> shall act as if the <b>-i</b> option were in |
| 28308 |                                                                                                       | effect for each file with invalid file name or link name values,                            |
| 28309 |                                                                                                       | allowing the user to provide a replacement name interactively.                              |
| 28310 |                                                                                                       | In <b>list</b> mode, <i>pax</i> shall behave identically to the <b>bypass</b> action.       |
| 28311 | <b>UTF-8</b>                                                                                          | When used in <b>read</b> , <b>copy</b> , or <b>list</b> mode and a file name, link          |
| 28312 |                                                                                                       | name, owner name, or any other field in an extended header                                  |
| 28313 |                                                                                                       | record cannot be translated from the <i>pax</i> UTF-8 codeset format                        |
| 28314 |                                                                                                       | to the codeset and current locale of the implementation, <i>pax</i>                         |
| 28315 |                                                                                                       | shall use the actual UTF-8 encoding for the name.                                           |
| 28316 | <b>write</b>                                                                                          | In <b>read</b> or <b>copy</b> mode, <i>pax</i> shall write the file, translating or         |
| 28317 |                                                                                                       | truncating the name, regardless of whether this may overwrite                               |
| 28318 |                                                                                                       | an existing file with a valid name. In <b>list</b> mode, <i>pax</i> shall behave            |
| 28319 |                                                                                                       | identically to the <b>bypass</b> action.                                                    |
| 28320 |                                                                                                       | If no <b>-o invalid=</b> option is specified, <i>pax</i> shall act as if                    |
| 28321 |                                                                                                       | <b>-o invalid=bypass</b> were specified. Any overwriting of existing                        |
| 28322 |                                                                                                       | files that may be allowed by the <b>-o invalid=</b> actions shall be                        |
| 28323 |                                                                                                       | subject to permission ( <b>-p</b> ) and modification time ( <b>-u</b> )                     |
| 28324 |                                                                                                       | restrictions, and shall be suppressed if the <b>-k</b> option is also                       |

|       |                       |                                                                                                                         |
|-------|-----------------------|-------------------------------------------------------------------------------------------------------------------------|
| 28325 |                       | specified.                                                                                                              |
| 28326 | <b>linkdata</b>       | (Applicable only to the <b>-x pax</b> format.) In <b>write</b> mode, <i>pax</i> shall                                   |
| 28327 |                       | write the contents of a file to the archive even when that file is                                                      |
| 28328 |                       | merely a hard link to a file whose contents have already been                                                           |
| 28329 |                       | written to the archive.                                                                                                 |
| 28330 | <b>listopt=format</b> |                                                                                                                         |
| 28331 |                       | This keyword specifies the output format of the table of contents produced                                              |
| 28332 |                       | when the <b>-v</b> option is specified in <b>list</b> mode. See <b>List Mode Format</b>                                 |
| 28333 |                       | <b>Specifications</b> on page 745. To avoid ambiguity, the <b>listopt=format</b> shall be                               |
| 28334 |                       | the only or final <b>keyword=value</b> pair in a <b>-o</b> option-argument; all characters in                           |
| 28335 |                       | the remainder of the option-argument shall be considered part of the format                                             |
| 28336 |                       | string. When multiple <b>-olistopt=format</b> options are specified, the format                                         |
| 28337 |                       | strings shall be considered a single, concatenated string, evaluated in                                                 |
| 28338 |                       | command line order.                                                                                                     |
| 28339 | <b>times</b>          |                                                                                                                         |
| 28340 |                       | (Applicable only to the <b>-x pax</b> format.) When used in <b>write</b> or <b>copy</b> mode, <i>pax</i>                |
| 28341 |                       | shall include <b>atime</b> , <b>ctime</b> , and <b>mtime</b> extended header records for each file.                     |
| 28342 |                       | See <b>pax Extended Header File Times</b> on page 753.                                                                  |
| 28343 |                       | In addition to these keywords, if the <b>-x pax</b> format is specified, any of the                                     |
| 28344 |                       | keywords and values defined in <b>pax Extended Header</b> on page 750, including                                        |
| 28345 |                       | implementation extensions, can be used in <b>-o</b> option-arguments, in either of two                                  |
| 28346 |                       | modes:                                                                                                                  |
| 28347 | <b>keyword=value</b>  |                                                                                                                         |
| 28348 |                       | When used in <b>write</b> or <b>copy</b> mode, these keyword/value pairs shall be                                       |
| 28349 |                       | included at the beginning of the archive as <b>typeflag g</b> global extended header                                    |
| 28350 |                       | records. When used in <b>read</b> or <b>list</b> mode, these keyword/value pairs shall act                              |
| 28351 |                       | as if they had been at the beginning of the archive as <b>typeflag g</b> global                                         |
| 28352 |                       | extended header records.                                                                                                |
| 28353 | <b>keyword:=value</b> |                                                                                                                         |
| 28354 |                       | When used in <b>write</b> or <b>copy</b> mode, these keyword/value pairs shall be                                       |
| 28355 |                       | included as records at the beginning of a <b>typeflag x</b> extended header for each                                    |
| 28356 |                       | file. (This is equivalent to the equal-sign form except that it creates no                                              |
| 28357 |                       | <b>typeflag g</b> global extended header records.) When used in <b>read</b> or <b>list</b> mode,                        |
| 28358 |                       | these keyword/value pairs shall act as if they were included as records at the                                          |
| 28359 |                       | end of each extended header; thus, they shall override any global or file-                                              |
| 28360 |                       | specific extended header record keywords of the same names. For example, in                                             |
| 28361 |                       | the command:                                                                                                            |
| 28362 |                       | <code>pax -r -o "</code>                                                                                                |
| 28363 |                       | <code>gname:=mygroup,</code>                                                                                            |
| 28364 |                       | <code>" &lt;archive</code>                                                                                              |
| 28365 |                       | the group name will be forced to a new value for all files read from the                                                |
| 28366 |                       | archive.                                                                                                                |
| 28367 |                       | The precedences of <b>-o</b> keywords over various fields in the archive are described in                               |
| 28368 |                       | <b>pax Extended Header Keyword Precedence</b> on page 752.                                                              |
| 28369 | <b>-p string</b>      | Specify one or more file characteristic options (privileges). The <i>string</i> option-                                 |
| 28370 |                       | argument shall be a string specifying file characteristics to be retained or discarded                                  |
| 28371 |                       | on extraction. The string shall consist of the specification characters <b>a</b> , <b>e</b> , <b>m</b> , <b>o</b> , and |

28372 **p.** Other implementation-dependent characters can be included. Multiple  
 28373 characteristics can be concatenated within the same string and multiple **-p** options  
 28374 can be specified. The meaning of the specification characters are as follows:

28375 **a** Do not preserve file access times.

28376 **e** Preserve the user ID, group ID, file mode bits (see the System Interface  
 28377 Definitions volume of IEEE Std. 1003.1-200x, Section 3.173, File Mode Bits),  
 28378 access time, modification time, and any other implementation-dependent file  
 28379 characteristics.

28380 **m** Do not preserve file modification times.

28381 **o** Preserve the user ID and group ID.

28382 **p** Preserve the file mode bits. Other implementation-dependent file mode  
 28383 attributes may be preserved.

28384 In the preceding list, “preserve” indicates that an attribute stored in the archive  
 28385 shall be given to the extracted file, subject to the permissions of the invoking  
 28386 process. The access and modification times of the file shall be preserved unless  
 28387 otherwise specified with the **-p** option or not stored in the archive. All attributes  
 28388 that are not preserved shall be determined as part of the normal file creation action  
 28389 (see Section 1.7.1.4 on page 11).

28390 If neither the **e** nor the **o** specification character is specified, or the user ID and  
 28391 group ID are not preserved for any reason, *pax* shall not set the S\_ISUID and  
 28392 S\_ISGID bits of the file mode.

28393 If the preservation of any of these items fails for any reason, *pax* shall write a  
 28394 diagnostic message to standard error. Failure to preserve these items shall affect  
 28395 the final exit status, but shall not cause the extracted file to be deleted.

28396 If file characteristic letters in any of the *string* option-arguments are duplicated or  
 28397 conflict with each other, the ones given last shall take precedence. For example, if  
 28398 **-p eme** is specified, file modification times are preserved.

28399 **-s replstr** Modify file or archive member names named by *pattern* or *file* operands according  
 28400 to the substitution expression *replstr*, using the syntax of the *ed* utility. The  
 28401 concepts of “address” and “line” are meaningless in the context of the *pax* utility,  
 28402 and shall not be supplied. The format shall be:

28403 **-s /old/new/[gp]**

28404 where as in *ed*, *old* is a basic regular expression and *new* can contain an ampersand,  
 28405 ‘\n’ (where *n* is a digit), backreferences, or subexpression matching. The *old*  
 28406 string also shall be permitted to contain <newline> characters.

28407 Any non-null character can be used as a delimiter (‘/’ shown here). Multiple **-s**  
 28408 expressions can be specified; the expressions shall be applied in the order  
 28409 specified, terminating with the first successful substitution. The optional trailing  
 28410 ‘g’ is as defined in the *ed* utility. The optional trailing ‘p’ shall cause successful  
 28411 substitutions to be written to standard error. File or archive member names that  
 28412 substitute to the empty string shall be ignored when reading and writing archives.

28413 **-t** Cause the access times of the archived files to be the same as they were before  
 28414 being read by *pax*.

28415 **-u** Ignore files that are older (having a less recent file modification time) than a pre-  
 28416 existing file or archive member with the same name. In **read** mode, an archive

|       |                  |                                                                                                                                                 |
|-------|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------|
| 28417 |                  | member with the same name as a file in the file system shall be extracted if the                                                                |
| 28418 |                  | archive member is newer than the file. In <b>write</b> mode, an archive file member with                                                        |
| 28419 |                  | the same name as a file in the file system shall be superseded if the file is newer                                                             |
| 28420 |                  | than the archive member. If <b>-a</b> is also specified, this is accomplished by appending                                                      |
| 28421 |                  | to the archive; otherwise, it is unspecified whether this is accomplished by actual                                                             |
| 28422 |                  | replacement in the archive or by appending to the archive. In <b>copy</b> mode, the file in                                                     |
| 28423 |                  | the destination hierarchy shall be replaced by the file in the source hierarchy or by                                                           |
| 28424 |                  | a link to the file in the source hierarchy if the file in the source hierarchy is newer.                                                        |
| 28425 | <b>-v</b>        | In <b>list</b> mode, produce a verbose table of contents (see the STDOUT section).                                                              |
| 28426 |                  | Otherwise, write archive member path names to standard error (see the STDERR                                                                    |
| 28427 |                  | section).                                                                                                                                       |
| 28428 | <b>-x format</b> | Specify the output archive format. The <i>pax</i> utility shall support the following                                                           |
| 28429 |                  | formats:                                                                                                                                        |
| 28430 | <b>cpio</b>      | The <i>cpio</i> interchange format; see the EXTENDED DESCRIPTION                                                                                |
| 28431 |                  | section. The default <i>blocksize</i> for this format for character special                                                                     |
| 28432 |                  | archive files shall be 5120. Implementations shall support all                                                                                  |
| 28433 |                  | <i>blocksize</i> values less than or equal to 32 256 that are multiples of 512.                                                                 |
| 28434 | <b>pax</b>       | The <i>pax</i> interchange format; see the EXTENDED DESCRIPTION                                                                                 |
| 28435 |                  | section. The default <i>blocksize</i> for this format for character special                                                                     |
| 28436 |                  | archive files shall be 5120. Implementations shall support all                                                                                  |
| 28437 |                  | <i>blocksize</i> values less than or equal to 32 256 that are multiples of 512.                                                                 |
| 28438 | <b>ustar</b>     | The <i>tar</i> interchange format; see the EXTENDED DESCRIPTION                                                                                 |
| 28439 |                  | section. The default <i>blocksize</i> for this format for character special                                                                     |
| 28440 |                  | archive files shall be 10 240. Implementations shall support all                                                                                |
| 28441 |                  | <i>blocksize</i> values less than or equal to 32 256 that are multiples of 512.                                                                 |
| 28442 |                  | Implementation-dependent formats shall specify a default block size as well as any                                                              |
| 28443 |                  | other block sizes supported for character special archive files.                                                                                |
| 28444 |                  | Any attempt to append to an archive file in a format different from the existing                                                                |
| 28445 |                  | archive format shall cause <i>pax</i> to exit immediately with a non-zero exit status.                                                          |
| 28446 |                  | In <b>copy</b> mode, if no <b>-x</b> format is specified, <i>pax</i> shall behave as if <b>-xpax</b> were                                       |
| 28447 |                  | specified.                                                                                                                                      |
| 28448 | <b>-X</b>        | When traversing the file hierarchy specified by a path name, <i>pax</i> shall not descend                                                       |
| 28449 |                  | into directories that have a different device ID ( <i>st_dev</i> ; see the System Interfaces                                                    |
| 28450 |                  | volume of IEEE Std. 1003.1-200x, <i>stat()</i> ).                                                                                               |
| 28451 |                  | The options that operate on the names of files or archive members ( <b>-c</b> , <b>-i</b> , <b>-n</b> , <b>-s</b> , <b>-u</b> , and <b>-v</b> ) |
| 28452 |                  | shall interact as follows. In <b>read</b> mode, the archive members shall be selected based on the user-                                        |
| 28453 |                  | specified <i>pattern</i> operands as modified by the <b>-c</b> , <b>-n</b> , and <b>-u</b> options. Then, any <b>-s</b> and <b>-i</b> options   |
| 28454 |                  | shall modify, in that order, the names of the selected files. The <b>-v</b> option shall write names                                            |
| 28455 |                  | resulting from these modifications.                                                                                                             |
| 28456 |                  | In <b>write</b> mode, the files shall be selected based on the user-specified path names as modified by                                         |
| 28457 |                  | the <b>-n</b> and <b>-u</b> options. Then, any <b>-s</b> and <b>-i</b> options shall modify, in that order, the names of                        |
| 28458 |                  | these selected files. The <b>-v</b> option shall write names resulting from these modifications.                                                |
| 28459 |                  | If both the <b>-u</b> and <b>-n</b> options are specified, <i>pax</i> shall not consider a file selected unless it is newer                     |
| 28460 |                  | than the file to which it is compared.                                                                                                          |

## 28461 List Mode Format Specifications

28462 In **list** mode with the **-o listopt=format** option, the *format* argument shall be applied for each  
 28463 selected file. The *pax* utility shall append a <newline> character to the **listopt** output for each  
 28464 selected file. The *format* argument shall be used as the *format* string described in the System  
 28465 Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 5, File Format Notation, with the  
 28466 exceptions 1. through 5. defined in the EXTENDED DESCRIPTION section of *printf*, plus the  
 28467 following exceptions:

28468 6. The sequence (*keyword*) can occur before a format conversion specifier. The conversion  
 28469 argument is defined by the value of *keyword*. The implementation shall support the  
 28470 following keywords:

28471 — Any of the Field Name entries in Table 4-13 on page 753 and Table 4-15 on page 757. The  
 28472 implementation may support the *cpio* keywords without the leading **c\_** in addition to the  
 28473 form required by Table 4-16 on page 758.

28474 — Any keyword defined for the extended header in **pax Extended Header** on page 750.

28475 — Any keyword provided as an implementation-dependent extension within the extended  
 28476 header defined in **pax Extended Header** on page 750.

28477 For example, the sequence "%(charset)s" is the string value of the name of the character  
 28478 set in the extended header.

28479 The result of the keyword conversion argument shall be the value from the applicable  
 28480 header field or extended header, without any trailing NULs.

28481 All keyword values used as conversion arguments shall be translated from the UTF-8  
 28482 encoding to the character set appropriate for the local file system, user database, and so on,  
 28483 as applicable.

28484 7. An additional conversion character, **T**, shall be used to specify time formats. The **T**  
 28485 conversion character can be preceded by the sequence (*keyword=subformat*), where *subformat*  
 28486 is a date format as defined by *date* operands. The default *keyword* shall be **mtime** and the  
 28487 default subformat shall be:

28488 %b %e %H:%M %Y

28489 8. An additional conversion character, **M**, shall be used to specify the file mode string as  
 28490 defined in *ls* Standard Output. If (*keyword*) is omitted, the **mode** keyword shall be used. For  
 28491 example, **%1M** writes the single character corresponding to the <entry type> field of the *ls*  
 28492 **-l** command.

28493 9. An additional conversion character, **D**, shall be used to specify the device for block or  
 28494 special files, if applicable, in an implementation-dependent format. If not applicable, and  
 28495 (*keyword*) is specified, then this conversion shall be equivalent to %(*keyword*)**u**. If not  
 28496 applicable, and (*keyword*) is omitted, then this conversion shall be equivalent to <space>.

28497 10. An additional conversion character, **F**, shall be used to specify a path name. The **F**  
 28498 conversion character can be preceded by a sequence of comma-separated keywords:

28499 (*keyword*[ ,*keyword*] . . . )

28500 The values for all the keywords that are non-null shall be concatenated together, each  
 28501 separated by a ' / '. The default shall be (**path**) if the keyword **path** is defined; otherwise,  
 28502 the default shall be (**prefix,name**).

28503 11. An additional conversion character, **L**, shall be used to specify a symbolic line expansion. If  
 28504 the current file is a symbolic link, then **%L** shall expand to:

|       |                                                                                                                     |  |
|-------|---------------------------------------------------------------------------------------------------------------------|--|
| 28505 | "%s -> %s", <value of keyword>, <contents of link>                                                                  |  |
| 28506 | Otherwise, the %L conversion character shall be the equivalent of %F.                                               |  |
| 28507 | <b>OPERANDS</b>                                                                                                     |  |
| 28508 | The following operands shall be supported:                                                                          |  |
| 28509 | <i>directory</i> The destination directory path name for <b>copy</b> mode.                                          |  |
| 28510 | <i>file</i> A path name of a file to be copied or archived.                                                         |  |
| 28511 | <i>pattern</i> A pattern matching one or more path names of archive members. A pattern must                         |  |
| 28512 | be given in the name-generating notation of the pattern matching notation in                                        |  |
| 28513 | Section 2.13 on page 92, including the file name expansion rules in Section 2.13.3                                  |  |
| 28514 | on page 94. The default, if no <i>pattern</i> is specified, is to select all members in the                         |  |
| 28515 | archive.                                                                                                            |  |
| 28516 | <b>STDIN</b>                                                                                                        |  |
| 28517 | In <b>write</b> mode, the standard input shall be used only if no <i>file</i> operands are specified. It shall be a |  |
| 28518 | text file containing a list of path names, one per line, without leading or trailing <blank>                        |  |
| 28519 | characters.                                                                                                         |  |
| 28520 | In <b>list</b> and <b>read</b> modes, if <b>-f</b> is not specified, the standard input shall be an archive file.   |  |
| 28521 | Otherwise, the standard input shall not be used.                                                                    |  |
| 28522 | <b>INPUT FILES</b>                                                                                                  |  |
| 28523 | The input file named by the <i>archive</i> option-argument, or standard input when the archive is read              |  |
| 28524 | from there, shall be a file formatted according to one of the specifications in the EXTENDED                        |  |
| 28525 | DESCRIPTION section or some other implementation-dependent format.                                                  |  |
| 28526 | The file <b>/dev/tty</b> shall be used to write prompts and read responses.                                         |  |
| 28527 | <b>ENVIRONMENT VARIABLES</b>                                                                                        |  |
| 28528 | The following environment variables shall affect the execution of <i>pax</i> :                                      |  |
| 28529 | <i>LANG</i> Provide a default value for the internationalization variables that are unset or null.                  |  |
| 28530 | If <i>LANG</i> is unset or null, the corresponding value from the implementation-                                   |  |
| 28531 | dependent default locale shall be used. If any of the internationalization variables                                |  |
| 28532 | contains an invalid setting, the utility shall behave as if none of the variables had                               |  |
| 28533 | been defined.                                                                                                       |  |
| 28534 | <i>LC_ALL</i> If set to a non-empty string value, override the values of all the other                              |  |
| 28535 | internationalization variables.                                                                                     |  |
| 28536 | <i>LC_COLLATE</i>                                                                                                   |  |
| 28537 | Determine the locale for the behavior of ranges, equivalence classes and multi-                                     |  |
| 28538 | character collating elements used in the pattern matching expressions for the                                       |  |
| 28539 | <i>pattern</i> operand, the basic regular expression for the <b>-s</b> option, and the extended                     |  |
| 28540 | regular expression defined for the <b>yesexpr</b> locale keyword in the <i>LC_MESSAGES</i>                          |  |
| 28541 | category.                                                                                                           |  |
| 28542 | <i>LC_CTYPE</i> Determine the locale for the interpretation of sequences of bytes of text data as                   |  |
| 28543 | characters (for example, single-byte as opposed to multi-byte characters in                                         |  |
| 28544 | arguments and input files), the behavior of character classes used in the extended                                  |  |
| 28545 | regular expression defined for the <b>yesexpr</b> locale keyword in the <i>LC_MESSAGES</i>                          |  |
| 28546 | category, and pattern matching.                                                                                     |  |
| 28547 | <i>LC_MESSAGES</i>                                                                                                  |  |
| 28548 | Determine the locale for the processing of affirmative responses that should be                                     |  |

28549                   used to affect the format and contents of diagnostic messages written to standard  
28550                   error.

28551           **LC\_TIME**   Determine the format and contents of date and time strings when the **-v** option is  
28552                   specified.

28553 **XSI**       **NLSPATH**   Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

28554           **TMPDIR**   Determine the path name that provides part of the default global extended header  
28555                   record file, as described for the **-o globexthdr=** keyword as described in the  
28556                   OPTIONS section.

28557 **ASYNCHRONOUS EVENTS**

28558           Default.

28559 **STDOUT**

28560           In **write** mode, if **-f** is not specified, the standard output shall be the archive formatted  
28561           according to one of the specifications in the EXTENDED DESCRIPTION section, or some other  
28562           implementation-dependent format (see **-x format**).

28563           In **list** mode, when the **-olistopt=format** has been specified, the selected archive members shall  
28564           be written to standard output using the format described under **List Mode Format**  
28565           **Specifications** on page 745. In **list** mode without the **-olistopt=format** option, the table of  
28566           contents of the the selected archive members shall be written to standard output using the  
28567           following format:

28568           "%s\n", <path name>

28569           If the **-v** option is specified in **list** mode, the table of contents of the selected archive members  
28570           shall be written to standard output using the following formats.

28571           For path names representing hard links to previous members of the archive:

28572           "%sΔ=Δ%s\n", <ls -l listing>, <linkname>

28573           For all other path names:

28574           "%s\n", <ls -l listing>

28575           where <ls -l listing> shall be the format specified by the **ls** utility with the **-l** option. When  
28576           writing path names in this format, it is unspecified what is written for fields for which the  
28577           underlying archive format does not have the correct information, although the correct number of  
28578           <blank> character-separated fields shall be written.

28579           In **list** mode, standard output shall not be buffered more than a line at a time.

28580 **STDERR**

28581           If **-v** is specified in **read**, **write**, or **copy** modes, **pax** shall write the path names it processes to the  
28582           standard error output using the following format:

28583           "%s\n", <path name>

28584           These path names shall be written as soon as processing is begun on the file or archive member,  
28585           and shall be flushed to standard error. The trailing <newline> character, which shall not be  
28586           buffered, is written when the file has been read or written.

28587           If the **-s** option is specified, and the replacement string has a trailing 'p', substitutions shall be  
28588           written to standard error in the following format:

28589           "%sΔ>>Δ%s\n", <original path name>, <new path name>

28590 In all operating modes of *pax*, optional messages of unspecified format concerning the input  
 28591 archive format and volume number, the number of files, blocks, volumes, and media parts as  
 28592 well as other diagnostic messages may be written to standard error.

28593 In all formats, for both standard output and standard error, it is unspecified how non-printable  
 28594 characters in path names or link names are written.

28595 When *pax* is in **read** mode or **list** mode, using the **-xpax** archive format, and a file name, link  
 28596 name, owner name, or any other field in an extended header record cannot be translated from  
 28597 the *pax* UTF-8 codeset format to the codeset and current locale of the implementation, *pax* shall  
 28598 write a diagnostic message to standard error, shall process the file as described for the **-o**  
 28599 **invalid=option**, and then shall process the next file in the archive.

## 28600 OUTPUT FILES

28601 In **read** mode, the extracted output files shall be of the archived file type. In **copy** mode, the  
 28602 copied output files shall be the type of the file being copied. In either mode, existing files in the  
 28603 destination hierarchy shall be overwritten only when all permission (**-p**), modification time (**-u**),  
 28604 and invalid-value (**-oinvalid=**) tests allow it.

28605 In **write** mode, the output file named by the **-f** option-argument shall be a file formatted  
 28606 according to one of the specifications in the EXTENDED DESCRIPTION section, or some other  
 28607 implementation-dependent format.

## 28608 EXTENDED DESCRIPTION

### 28609 **pax Interchange Format**

28610 A *pax* archive tape or file produced in the **-xpax** format shall contain a series of blocks. The  
 28611 physical layout of the archive shall be identical to the **ustar** format described in **ustar**  
 28612 **Interchange Format** on page 753. Each file archived shall be represented by the following  
 28613 sequence:

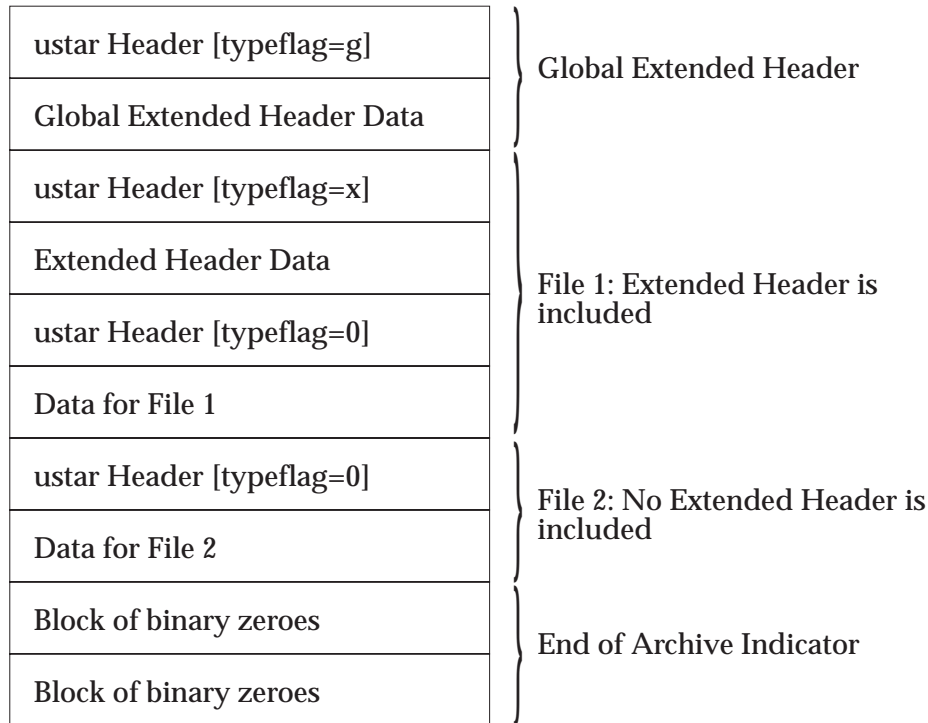
- 28614 • An optional header block with extended header records. This header block is of the form  
 28615 described in **pax Header Block** on page 749, with a *typeflag* value of **x** or **g**. The extended  
 28616 header records, described in **pax Extended Header** on page 750, are included as the data for  
 28617 this header block.
- 28618 • A header block that describes the file. Any fields in the preceding optional extended header  
 28619 override the associated fields in this header block for this file.
- 28620 • Zero or more blocks that contain the contents of the file.

28621 At the end of the archive file there shall be two 512-byte blocks filled with binary zeroes,  
 28622 interpreted as an end-of-archive indicator.

28623 A schematic of an example archive with global extended header records and two actual files is  
 28624 shown in Figure 4-1 on page 749. In the example, the second file in the archive has no extended  
 28625 header preceding it, presumably because it has no need for extended attributes.



28626



28627

Figure 4-1 pax Format Archive Example

28628

**pax Header Block**

28629

The *pax* header block shall be identical to the **ustar** header block described in **ustar Interchange Format** on page 753, except that two additional *typeflag* values are defined:

28630

28631

**x** Represents extended header records for the following file in the archive (which shall have its own **ustar** header block). The format of these extended header records shall be as described in **pax Extended Header** on page 750.

28632

28633

28634

**g** Represents global extended header records for the following files in the archive. The format of these extended header records shall be as described in **pax Extended Header** on page 750. Each value shall affect all subsequent files that do not override that value in their own extended header record and until another global extended header record is reached that provides another value for the same field. The *typeflag g* global headers should not be used with interchange media that could suffer partial data loss in transporting the archive.

28635

28636

28637

28638

28639

28640

For both of these types, the *size* field shall be the size of the extended header records in octets. The other fields in the header block are not meaningful to this version of the *pax* utility. However, if this archive is read by a *pax* utility conforming to a previous version of IEEE Std. 1003.1-200x, the header block fields are used to create a regular file that contains the extended header records as data. Therefore, header block field values should be selected to provide reasonable file access to this regular file.

28641

28642

28643

28644

28645

28646

A further difference from the **ustar** header block is that data blocks for files of *typeflag 1* (hard link) may be included, which means that the *size* field may be greater than zero. Archives created by **pax -o linkdata** shall include these data blocks with the hard links.

28647

28648

28649 **pax Extended Header**

28650 A *pax* extended header contains values that are inappropriate for the **ustar** header block because  
 28651 of limitations in that format: fields requiring a character encoding other than that described in  
 28652 the ISO/IEC 646:1991 standard, fields representing file attributes not described in the **ustar**  
 28653 header, and fields whose format or length do not fit the requirements of the **ustar** header. The  
 28654 values in an extended header add attributes to the following file (or files; see the description of  
 28655 the *typeflag g* header block) or override values in the following header block(s), as indicated in  
 28656 the following list of keywords.

28657 An extended header shall consist of one or more records, each constructed as follows:

28658 "%d %s=%s\n", <length>, <keyword>, <value>

28659 The extended header records shall be encoded according to the ISO/IEC 10646-1:1993 standard  
 28660 (UTF-8). The <length> field, <blank> character, equals sign, and <newline> character shown  
 28661 shall be limited to the portable character set, as encoded in UTF-8. The <keyword> and <value>  
 28662 fields can be any UTF-8 characters. The <length> field shall be the decimal length of the extended  
 28663 header record in octets, including the trailing <newline> character.

28664 The <keyword> field shall be one of the entries from the following list or a keyword provided as  
 28665 an implementation extension. Keywords consisting entirely of lowercase letters, digits, and  
 28666 periods are reserved for future standardization. A keyword shall not include an equals sign. (In  
 28667 the following list, the notations “file(s)” or “block(s)” is used to acknowledge that a keyword  
 28668 affects the following single file after a *typeflag x* extended header, but possibly multiple files after  
 28669 *typeflag g*. Any requirements in the list for *pax* to include a record when in **write** or **copy** mode  
 28670 shall apply only when such a record has not already been provided through the use of the **-o**  
 28671 option. When used in **copy** mode, *pax* shall behave as if an archive had been created with  
 28672 applicable extended header records and then extracted.)

28673 **atime** The file access time for the following file(s), equivalent to the value of the *st\_atime*  
 28674 member of the **stat** structure for a file, as described by the *stat()* function. The  
 28675 access time shall be restored if the process has the appropriate privilege required  
 28676 to do so. The format of the <value> shall be as described in **pax Extended Header**  
 28677 **File Times** on page 753.

28678 **charset** The name of the character set used to encode the data in the following file(s). The  
 28679 entries in the following table are defined to refer to known standards; additional  
 28680 names may be agreed on between the originator and recipient.

| <value>                 | Formal Standard               |
|-------------------------|-------------------------------|
| ISO-IRA646Δ1990         | ISO/IEC 646:1990              |
| ISO-IRA8859Δ1Δ1987      | ISO/IEC 8859-1:1987           |
| ISO-IRA8859Δ2Δ1987      | ISO/IEC 8859-2:1987           |
| ISO-IRA10646Δ1993       | ISO/IEC 10646:1993            |
| ISO-IRA10646Δ1993ΔUTF-8 | ISO/IEC 10646, UTF-8 encoding |
| BINARY                  | None.                         |

28688 The encoding is included in an extended header for information only; when *pax* is  
 28689 used as described in IEEE Std. 1003.1-200x, it shall not translate the file data into  
 28690 any other encoding. The **BINARY** entry indicates unencoded binary data.

28691 When used in **write** or **copy** mode, it is implementation-dependent whether *pax*  
 28692 includes a **charset** extended header record for a file.

28693 **comment** A series of characters used as a comment. All characters in the <value> field shall  
 28694 be ignored by *pax*.

|       |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 28695 | <b>ctime</b>        | The file creation time for the following file(s), equivalent to the value of the <i>st_ctime</i> member of the <b>stat</b> structure for a file, as described by the <i>stat()</i> function. The creation time shall be restored if the process has the appropriate privilege required to do so. The format of the <value> shall be as described in <b>pax Extended Header File Times</b> on page 753.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 28696 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28697 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28698 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28699 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28700 | <b>gid</b>          | The group ID of the group that owns the file, expressed as a decimal number using digits from the ISO/IEC 646:1991 standard. This record shall override the <i>gid</i> field in the following header block(s). When used in <b>write</b> or <b>copy</b> mode, <i>pax</i> shall include a <i>gid</i> extended header record for each file whose group ID is greater than 99 999 999.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28701 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28702 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28703 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28704 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28705 | <b>gname</b>        | The group of the file(s), formatted as a group name in the group database. This record shall override the <i>gid</i> and <i>gname</i> fields in the following header block(s), and any <i>gid</i> extended header record. When used in <b>read</b> , <b>copy</b> , or <b>list</b> mode, <i>pax</i> shall translate the name from the UTF-8 encoding in the header record to the character set appropriate for the group database on the receiving system. If any of the UTF-8 characters cannot be translated, and if the <b>-oinvalid=</b> UTF-8 option is not specified, the results are implementation-dependent. When used in <b>write</b> or <b>copy</b> mode, <i>pax</i> shall include a <b>gname</b> extended header record for each file whose group name cannot be represented entirely with the letters and digits of the portable character set.                                                                |
| 28706 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28707 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28708 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28709 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28710 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28711 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28712 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28713 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28714 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28715 | <b>linkpath</b>     | The path name of a link being created to another file, of any type, previously archived. This record shall override the <i>linkname</i> field in the following <b>ustar</b> header block(s). The following <b>ustar</b> header block shall determine the type of link created. If <i>typeflag</i> of the following header block is 1, it shall be a hard link. If <i>typeflag</i> is 2, it shall be a symbolic link and the <b>linkpath</b> value shall be the contents of the symbolic link. The <i>pax</i> utility shall translate the name of the link (contents of the symbolic link) from the UTF-8 encoding to the character set appropriate for the local file system. When used in <b>write</b> or <b>copy</b> mode, <i>pax</i> shall include a <b>linkpath</b> extended header record for each link whose path name cannot be represented entirely with the members of the portable character set other than NUL. |
| 28716 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28717 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28718 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28719 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28720 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28721 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28722 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28723 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28724 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28725 | <b>mtime</b>        | The file modification time of the following file(s), equivalent to the value of the <i>st_mtime</i> member of the <b>stat</b> structure for a file, as described in the <i>stat()</i> function. This record shall override the <i>mtime</i> field in the following header block(s). The modification time shall be restored if the process has the appropriate privilege required to do so. The format of the <value> shall be as described in <b>pax Extended Header File Times</b> on page 753.                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 28726 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28727 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28728 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28729 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28730 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28731 | <b>path</b>         | The path name of the following file(s). This record shall override the <i>name</i> and <i>prefix</i> fields in the following header block(s). The <i>pax</i> utility shall translate the path name of the file from the UTF-8 encoding to the character set appropriate for the local file system.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 28732 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28733 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28734 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28735 |                     | When used in <b>write</b> or <b>copy</b> mode, <i>pax</i> shall include a <i>path</i> extended header record for each file whose path name cannot be represented entirely with the members of the portable character set other than NUL.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 28736 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28737 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28738 | <b>realtime.any</b> | The keywords prefixed by “realtime.” are reserved for future standardization.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 28739 | <b>security.any</b> | The keywords prefixed by “security.” are reserved for future standardization.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 28740 | <b>size</b>         | The size of the file in octets, expressed as a decimal number using digits from the ISO/IEC 646:1991 standard. This record shall override the <i>size</i> field in the following header block(s). When used in <b>write</b> or <b>copy</b> mode, <i>pax</i> shall include a                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 28741 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 28742 |                     |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |

|       |                                               |                                                                                                                       |
|-------|-----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| 28743 |                                               | <i>size</i> extended header record for each file with a size value greater than                                       |
| 28744 |                                               | 999 999 999 999.                                                                                                      |
| 28745 | <b>uid</b>                                    | The user ID of the file owner, expressed as a decimal number using digits from the                                    |
| 28746 |                                               | ISO/IEC 646:1991 standard. This record shall override the <i>uid</i> field in the                                     |
| 28747 |                                               | following header block(s). When used in <b>write</b> or <b>copy</b> mode, <i>pax</i> shall include a                  |
| 28748 |                                               | <i>uid</i> extended header record for each file whose owner ID is greater than 99 999 999.                            |
| 28749 | <b>uname</b>                                  | The owner of the following file(s), formatted as a user name in the user database.                                    |
| 28750 |                                               | This record shall override the <i>uid</i> and <i>uname</i> fields in the following header block(s),                   |
| 28751 |                                               | and any <i>uid</i> extended header record. When used in <b>read</b> , <b>copy</b> , or <b>list</b> mode, <i>pax</i>   |
| 28752 |                                               | shall translate the name from the UTF-8 encoding in the header record to the                                          |
| 28753 |                                               | character set appropriate for the user database on the receiving system. If any of                                    |
| 28754 |                                               | the UTF-8 characters cannot be translated, and if the <b>-oinvalid=</b> UTF-8 option is                               |
| 28755 |                                               | not specified, the results are implementation-dependent. When used in <b>write</b> or                                 |
| 28756 |                                               | <b>copy</b> mode, <i>pax</i> shall include a <b>uname</b> extended header record for each file whose                  |
| 28757 |                                               | user name cannot be represented entirely with the letters and digits of the portable                                  |
| 28758 |                                               | character set.                                                                                                        |
| 28759 |                                               | If the <i>&lt;value&gt;</i> field is zero length, it shall delete any header block field, previously entered          |
| 28760 |                                               | extended header value, or global extended header value of the same name.                                              |
| 28761 |                                               | If a keyword in an extended header record (or in a <b>-o</b> option-argument) overrides or deletes a                  |
| 28762 |                                               | corresponding field in the <b>ustar</b> header block, <i>pax</i> shall ignore the contents of that header block       |
| 28763 |                                               | field.                                                                                                                |
| 28764 |                                               | Unlike the <b>ustar</b> header block fields, NULs shall not delimit <i>&lt;value&gt;</i> s; all characters within the |
| 28765 |                                               | <i>&lt;value&gt;</i> field shall be considered data for the field. None of the length limitations of the <b>ustar</b> |
| 28766 |                                               | header block fields in Table 4-13 on page 753 shall apply to the extended header records.                             |
| 28767 | <b>pax Extended Header Keyword Precedence</b> |                                                                                                                       |
| 28768 |                                               | This section describes the precedence in which the various header records and fields and                              |
| 28769 |                                               | command line options are selected to apply to a file in the archive. When <i>pax</i> is used in <b>read</b> or        |
| 28770 |                                               | <b>list</b> modes, it shall determine a file attribute in the following sequence:                                     |
| 28771 | 1.                                            | If <b>-odelete=keyword-prefix</b> is used, the affected attributes shall be determined from step 7.,                  |
| 28772 |                                               | if applicable, or ignored otherwise.                                                                                  |
| 28773 | 2.                                            | If <b>-okeyword:=</b> is used, the affected attributes shall be ignored.                                              |
| 28774 | 3.                                            | If <b>-okeyword:=value</b> is used, the affected attribute shall be assigned the value.                               |
| 28775 | 4.                                            | If there is a <i>typeflag</i> <b>x</b> extended header record, the affected attribute shall be assigned the           |
| 28776 |                                               | <i>&lt;value&gt;</i> . When extended header records conflict, the last one given in the header shall take             |
| 28777 |                                               | precedence.                                                                                                           |
| 28778 | 5.                                            | If <b>-okeyword=value</b> is used, the affected attribute shall be assigned the value.                                |
| 28779 | 6.                                            | If there is a <i>typeflag</i> <b>g</b> global extended header record, the affected attribute shall be                 |
| 28780 |                                               | assigned the <i>&lt;value&gt;</i> . When global extended header records conflict, the last one given in               |
| 28781 |                                               | the global header shall take precedence.                                                                              |
| 28782 | 7.                                            | Otherwise, the attribute shall be determined from the <b>ustar</b> header block.                                      |

### 28783 **pax Extended Header File Times**

28784 The *pax* utility shall write **atime** and **ctime** records for each file in **write** or **copy** modes only if  
 28785 the **-otimes** option is specified; *pax* shall write a **mtime** record for each file in **write** or **copy**  
 28786 modes if the file system of the underlying implementation supports time granularities smaller  
 28787 than that required by the **ustar** header block described in **ustar Interchange Format**. All of these  
 28788 time records shall be formatted as a decimal representation of the time in seconds since the  
 28789 Epoch. If a period ( ' . ' ) decimal point character is present, the digits to the right of the point  
 28790 shall represent the units of a subsecond timing granularity, where the first digit is tenths of a  
 28791 second and each subsequent digit is a tenth of the previous digit. Implementations may ignore  
 28792 any portion of the subsecond digits for which they do not support the necessary timing  
 28793 granularity; they shall not perform any rounding operation.

### 28794 **ustar Interchange Format**

28795 A **ustar** archive tape or file shall contain a series of blocks. Each block shall be a fixed-size block  
 28796 of 512 octets (see below). Although this format may be thought of as being stored on 9-track  
 28797 industry-standard 12.7mm (0.5in) magnetic tape, other types of transportable media are not  
 28798 excluded. Each file archived shall be represented by a header block that describes the file,  
 28799 followed by zero or more blocks that give the contents of the file. At the end of the archive file  
 28800 there shall be two 512-octet blocks filled with binary zeros, interpreted as an end-of-archive  
 28801 indicator.

28802 The blocks may be grouped for physical I/O operations, as described under the **-bblocksize** and  
 28803 **-x ustar** options. Each group of blocks may be written with a single operation equivalent to the  
 28804 *write()* function. On magnetic tape, the result of this write shall be a single tape record. The last  
 28805 group of blocks always shall be at the full size, so blocks after the two zero blocks may contain  
 28806 undefined data.

28807 The header block shall be structured as shown in the following table. All lengths and offsets are  
 28808 in decimal.

28809 **Table 4-13** ustar Header Block

| Field Name      | Octet Offset | Length (in Octets) |
|-----------------|--------------|--------------------|
| <i>name</i>     | 0            | 100                |
| <i>mode</i>     | 100          | 8                  |
| <i>uid</i>      | 108          | 8                  |
| <i>gid</i>      | 116          | 8                  |
| <i>size</i>     | 124          | 12                 |
| <i>mtime</i>    | 136          | 12                 |
| <i>chksum</i>   | 148          | 8                  |
| <i>typeflag</i> | 156          | 1                  |
| <i>linkname</i> | 157          | 100                |
| <i>magic</i>    | 257          | 6                  |
| <i>version</i>  | 263          | 2                  |
| <i>uname</i>    | 265          | 32                 |
| <i>gname</i>    | 297          | 32                 |
| <i>devmajor</i> | 329          | 8                  |
| <i>devminor</i> | 337          | 8                  |
| <i>prefix</i>   | 345          | 155                |

28827 All characters in the header block shall be represented in the coded character set of the  
 28828 ISO/IEC 646:1991 standard. For maximum portability between implementations, names should  
 28829 be selected from characters represented by the portable file name character set as octets with the

28830 most significant bit zero. If an implementation supports the use of characters outside of slash  
 28831 and the portable file name character set in names for files, users, and groups, one or more  
 28832 implementation-dependent encodings of these characters shall be provided for interchange  
 28833 purposes.

#### 28834 **Notes to Reviewers**

28835 *This section with side shading will not appear in the final copy. - Ed.*

28836 The following paragraph was in XCU and is not in 1003.2b  
 28837 However, the *pax* utility shall never create file names on the local system that cannot be accessed  
 28838 via the procedures described previously in this volume of IEEE Std. 1003.1-200x. If a file name is  
 28839 found on the medium that would create an invalid file name, it is implementation-dependent  
 28840 whether the data from the file is stored on the file hierarchy and under what name it is stored.  
 28841 The *pax* utility may choose to ignore these files as long as it produces an error indicating that the  
 28842 file is being ignored.

28843 Each field within the header block is contiguous; that is, there is no padding used. Each character  
 28844 on the archive medium shall be stored contiguously.

28845 The fields *magic*, *uname*, and *gname* are character strings each terminated by a NUL character.  
 28846 The fields *name*, *linkname*, and *prefix* are NUL-terminated character strings except when all  
 28847 characters in the array contain non-NUL characters including the last character. The *version* field  
 28848 is two octets containing the characters "00" (zero-zero). The *typeflag* contains a single character.  
 28849 All other fields are leading zero-filled octal numbers using digits from the ISO/IEC 646:1991  
 28850 standard IRV. Each numeric field is terminated by one or more <space> or NUL characters.

28851 The *name* and the *prefix* fields shall produce the path name of the file. A new path name shall be  
 28852 formed, if *prefix* is not an empty string (its first character is not NUL), by concatenating *prefix* (up  
 28853 to the first NUL character), a slash character, and *name*; otherwise, *name* is used alone. In either  
 28854 case, *name* is terminated at the first NUL character. If *prefix* begins with a NUL character, it shall  
 28855 be ignored. In this manner, path names of at most 256 characters can be supported. If a path  
 28856 name does not fit in the space provided, *pax* shall notify the user of the error, and shall not store  
 28857 any part of the file—header or data—on the medium.

28858 The *linkname* field, described below, shall not use the *prefix* to produce a path name. As such, a  
 28859 *linkname* is limited to 100 characters. If the name does not fit in the space provided, *pax* shall  
 28860 notify the user of the error, and shall not attempt to store the link on the medium.

28861 The *mode* field provides 12 bits encoded in the ISO/IEC 646:1991 standard octal digit  
 28862 representation. The encoded bits shall represent the following values:

Table 4-14 ustar mode Field

| Bit Value | IEEE Std. 1003.1-200x Bit | Description                                     |
|-----------|---------------------------|-------------------------------------------------|
| 04 000    | S_ISUID                   | Set UID on execution.                           |
| 02 000    | S_ISGID                   | Set GID on execution.                           |
| 01 000    | <reserved>                | Reserved for future standardization.            |
| 00 400    | S_IRUSR                   | Read permission for file owner class.           |
| 00 200    | S_IWUSR                   | Write permission for file owner class.          |
| 00 100    | S_IXUSR                   | Execute/search permission for file owner class. |
| 00 040    | S_IRGRP                   | Read permission for file group class.           |
| 00 020    | S_IWGRP                   | Write permission for file group class.          |
| 00 010    | S_IXGRP                   | Execute/search permission for file group class. |
| 00 004    | S_IROTH                   | Read permission for file other class.           |
| 00 002    | S_IWOTH                   | Write permission for file other class.          |
| 00 001    | S_IXOTH                   | Execute/search permission for file other class. |

When appropriate privilege is required to set one of these mode bits, and the user restoring the files from the archive does not have the appropriate privilege, the mode bits for which the user does not have appropriate privilege shall be ignored. Some of the mode bits in the archive format are not mentioned elsewhere in this volume of IEEE Std. 1003.1-200x. If the implementation does not support those bits, they may be ignored.

The *uid* and *gid* fields are the user and group ID of the owner and group of the file, respectively.

The *size* field is the size of the file in octets. If the *typeflag* field is set to specify a file to be of type 1 (a link) or 2 (reserved for symbolic links), the *size* field shall be specified as zero. If the *typeflag* field is set to specify a file of type 5 (directory), the *size* field shall be interpreted as described under the definition of that record type. No data blocks are stored for types 1, 2, or 5. If the *typeflag* field is set to 3 (character special file), 4 (block special file), or 6 (FIFO), the meaning of the *size* field is unspecified by this volume of IEEE Std. 1003.1-200x, and no data blocks shall be stored on the medium. Additionally, for 6, the *size* field shall be ignored when reading. If the *typeflag* field is set to any other value, the number of blocks written following the header shall be  $(size+511)/512$ , ignoring any fraction in the result of the division.

The *mtime* field shall be the modification time of the file at the time it was archived. It is the ISO/IEC 646:1991 standard representation of the octal value of the modification time obtained from the *stat()* function.

The *chksum* field shall be the ISO/IEC 646:1991 standard IRV representation of the octal value of the simple sum of all octets in the header block. Each octet in the header shall be treated as an unsigned value. These values shall be added to an unsigned integer, initialized to zero, the precision of which is not less than 17 bits. When calculating the checksum, the *chksum* field is treated as if it were all spaces.

The *typeflag* field specifies the type of file archived. If a particular implementation does not recognize the type, or the user does not have appropriate privilege to create that type, the file shall be extracted as if it were a regular file if the file type is defined to have a meaning for the *size* field that could cause data blocks to be written on the medium (see the previous description for *size*). If conversion to a regular file occurs, the *pax* utility shall produce an error indicating that the conversion took place. All of the *typeflag* fields shall be coded in the ISO/IEC 646:1991 standard IRV:

0 Represents a regular file. For backward compatibility, a *typeflag* value of binary zero ('`\0`') should be recognized as meaning a regular file when extracting files from the archive. Archives written with this version of the archive file format create regular files with a *typeflag* value of the ISO/IEC 646:1991 standard IRV '`0`'.

|       |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|-------|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 28911 | 1    | Represents a file linked to another file, of any type, previously archived. Such files are identified by each file having the same device and file serial number. The linked-to name is specified in the <i>linkname</i> field with a NUL-character terminator if it is less than 100 octets in length.                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 28912 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28913 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28914 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28915 | 2    | Represents a symbolic link. The contents of the symbolic link shall be stored in the <i>linkname</i> field.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 28916 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28917 | 3, 4 | Represent character special files and block special files respectively. In this case the <i>devmajor</i> and <i>devminor</i> fields shall contain information defining the device, the format of which is unspecified by this volume of IEEE Std. 1003.1-200x. Implementations may map the device specifications to their own local specification or may ignore the entry.                                                                                                                                                                                                                                                                                                                                                            |
| 28918 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28919 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28920 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28921 | 5    | Specifies a directory or subdirectory. On systems where disk allocation is performed on a directory basis, the <i>size</i> field shall contain the maximum number of octets (which may be rounded to the nearest disk block allocation unit) that the directory may hold. A <i>size</i> field of zero indicates no such limiting. Systems that do not support limiting in this manner should ignore the <i>size</i> field.                                                                                                                                                                                                                                                                                                            |
| 28922 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28923 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28924 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28925 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28926 | 6    | Specifies a FIFO special file. Note that the archiving of a FIFO file archives the existence of this file and not its contents.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28927 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28928 | 7    | Reserved to represent a file to which an implementation has associated some high-performance attribute. Implementations without such extensions should treat this file as a regular file (type 0).                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 28929 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28930 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28931 | A-Z  | The letters 'A' to 'Z', inclusive, are reserved for custom implementations. All other values are reserved for future revisions of IEEE Std. 1003.1-200x.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 28932 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28933 |      | The <i>magic</i> field is the specification that this archive was output in this archive format. If this field contains <b>ustar</b> (the five characters from the ISO/IEC 646:1991 standard IRV shown followed by NUL), the <i>uname</i> and <i>gname</i> fields shall contain the ISO/IEC 646:1991 standard IRV representation of the owner and group of the file, respectively (truncated to fit, if necessary). When the file is restored by a privileged, protection-preserving version of the utility, the user and group databases shall be scanned for these names. If found, the user and group IDs contained within these files shall be used rather than the values contained within the <i>uid</i> and <i>gid</i> fields. |
| 28934 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28935 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28936 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28937 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28938 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28939 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28940 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28941 |      | <b>cpio Interchange Format</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 28942 |      | The octet-oriented <i>cpio</i> archive format shall be a series of entries, each comprising a header that describes the file, the name of the file, and then the contents of the file.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 28943 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28944 |      | An archive may be recorded as a series of fixed-size blocks of octets. This blocking shall be used only to make physical I/O more efficient. The last group of blocks shall be always at the full size.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 28945 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28946 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 28947 |      | For the octet-oriented <i>cpio</i> archive format, the individual entry information shall be in the order indicated and described by the following table; see also the < <b>cpio.h</b> > header.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 28948 |      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |



Table 4-15 Octet-Oriented *cpio* Archive Entry

| Header Field Name    | Length (in Octets) | Interpreted as   |
|----------------------|--------------------|------------------|
| <i>c_magic</i>       | 6                  | Octal number     |
| <i>c_dev</i>         | 6                  | Octal number     |
| <i>c_ino</i>         | 6                  | Octal number     |
| <i>c_mode</i>        | 6                  | Octal number     |
| <i>c_uid</i>         | 6                  | Octal number     |
| <i>c_gid</i>         | 6                  | Octal number     |
| <i>c_nlink</i>       | 6                  | Octal number     |
| <i>c_rdev</i>        | 6                  | Octal number     |
| <i>c_mtime</i>       | 11                 | Octal number     |
| <i>c_namesize</i>    | 6                  | Octal number     |
| <i>c_filesize</i>    | 11                 | Octal number     |
| File Name Field Name | Length             | Interpreted as   |
| <i>c_name</i>        | <i>c_namesize</i>  | Path name string |
| File Data Field Name | Length             | Interpreted as   |
| <i>c_filedata</i>    | <i>c_filesize</i>  | Data             |

**cpio Header**

For each file in the archive, a header as defined previously shall be written. The information in the header fields is written as streams of the ISO/IEC 646: 1991 standard characters interpreted as octal numbers. The octal numbers shall be extended to the necessary length by appending the ISO/IEC 646: 1991 standard IRV zeros at the most-significant-digit end of the number; the result is written to the most-significant digit of the stream of octets first. The fields shall be interpreted as follows:

|                             |                                                                                                                                                                                                                                                       |
|-----------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <i>c_magic</i>              | Identify the archive as being a transportable archive by containing the identifying value "070707".                                                                                                                                                   |
| <i>c_dev</i> , <i>c_ino</i> | Contains values that uniquely identify the file within the archive (that is, no files contain the same pair of <i>c_dev</i> and <i>c_ino</i> values unless they are links to the same file). The values shall be determined in an unspecified manner. |
| <i>c_mode</i>               | Contains the file type and access permissions as defined in the following table.                                                                                                                                                                      |

Table 4-16 Values for *cpio c\_mode* Field

| File Permissions Name | Value    | Indicates              |
|-----------------------|----------|------------------------|
| C_IRUSR               | 000 400  | Read by owner          |
| C_IWUSR               | 000 200  | Write by owner         |
| C_IXUSR               | 000 100  | Execute by owner       |
| C_IRGRP               | 000 040  | Read by group          |
| C_IWGRP               | 000 020  | Write by group         |
| C_IXGRP               | 000 010  | Execute by group       |
| C_IROTH               | 000 004  | Read by others         |
| C_IWOTH               | 000 002  | Write by others        |
| C_IXOTH               | 000 001  | Execute by others      |
| C_ISUID               | 004 000  | Set <i>uid</i>         |
| C_ISGID               | 002 000  | Set <i>gid</i>         |
| C_ISVTX               | 001 000  | Reserved               |
| File Type Name        | Value    | Indicates              |
| C_ISDIR               | 040 000  | Directory              |
| C_ISFIFO              | 010 000  | FIFO                   |
| C_ISREG               | 0100 000 | Regular file           |
| C_ISBLK               | 060 000  | Block special file     |
| C_ISCHR               | 020 000  | Character special file |
| C_ISCTG               | 0110 000 | Reserved               |
| C_ISLNK               | 0120 000 | Reserved               |
| C_ISSOCK              | 0140 000 | Reserved               |

Directories, FIFOs, and regular files shall be supported on a system conforming to this volume of IEEE Std. 1003.1-200x; additional values defined previously are reserved for compatibility with existing systems. Additional file types may be supported; however, such files should not be written to archives intended to be transported to other systems.

|                   |                                                                                                                         |
|-------------------|-------------------------------------------------------------------------------------------------------------------------|
| <i>c_uid</i>      | Contains the user ID of the owner.                                                                                      |
| <i>c_gid</i>      | Contains the group ID of the group.                                                                                     |
| <i>c_nlink</i>    | Contains the number of links referencing the file at the time the archive was created.                                  |
| <i>c_rdev</i>     | Contains implementation-dependent information for character or block special files.                                     |
| <i>c_mtime</i>    | Contains the latest time of modification of the file at the time the archive was created.                               |
| <i>c_namesize</i> | Contains the length of the path name, including the terminating NUL character.                                          |
| <i>c_filesize</i> | Contains the length of the file in octets. This shall be the length of the data section following the header structure. |

## 29018 **cpio File Name**

29019 The `c_name` field shall contain the path name of the file. The length of this field in octets is the  
 29020 value of `c_namesize`.

## 29021 **Notes to Reviewers**

29022 *This section with side shading will not appear in the final copy. - Ed.*

29023 The following paragraph in XCU is not present in 1003.2b. Comments?

29024 If a file name is found on the medium that would create an invalid path name, it is  
 29025 implementation-dependent whether the data from the file is stored on the file hierarchy and  
 29026 under what name it is stored.

29027 All characters shall be represented in the ISO/IEC 646:1991 standard IRV. For maximum  
 29028 portability between implementations, names should be selected from characters represented by  
 29029 the portable file name character set as octets with the most significant bit zero. If an  
 29030 implementation supports the use of characters outside the portable file name character set in  
 29031 names for files, users, and groups, one or more implementation-dependent encodings of these  
 29032 characters shall be provided for interchange purposes.

## 29033 **Notes to Reviewers**

29034 *This section with side shading will not appear in the final copy. - Ed.*

29035 The next sentence is in XCU but not in 1003.2b

29036 However, the `pax` utility shall never create file names on the local system that cannot be accessed  
 29037 via the procedures described previously in this volume of IEEE Std. 1003.1-200x. If a file name is  
 29038 found on the medium that would create an invalid file name, it is implementation-dependent  
 29039 whether the data from the file is stored on the local file system and under what name it is stored.  
 29040 The `pax` utility may choose to ignore these files as long as it produces an error indicating that the  
 29041 file is being ignored.

## 29042 **cpio File Data**

29043 Following `c_name`, there shall be `c_filesiz` octets of data. Interpretation of such data occurs in a  
 29044 manner dependent on the file. If `c_filesiz` is zero, no data shall be contained in `c_filedata`.

## 29045 **Notes to Reviewers**

29046 *This section with side shading will not appear in the final copy. - Ed.*

29047 The following bullet items are in XCU but not in 1003.2b. Comments?

29048 When restoring from an archive:

- 29049 • If the user does not have the appropriate privilege to create a file of the specified type, `pax`  
 29050 shall ignore the entry and write an error message to standard error.
- 29051 • Only regular files have data to be restored. Presuming a regular file meets any selection  
 29052 criteria that might be imposed on the format-reading utility by the user, such data shall be  
 29053 restored.
- 29054 • If a user does not have appropriate privilege to set a particular mode flag, the flag shall be  
 29055 ignored. Some of the mode flags in the archive format are not mentioned elsewhere in this  
 29056 volume of IEEE Std. 1003.1-200x. If the implementation does not support those flags, they  
 29057 may be ignored.

29058 **cpio Special Entries**

29059 FIFO special files, directories, and the trailer shall be recorded with *c\_filesize* equal to zero. For  
 29060 other special files, *c\_filesize* is unspecified by this volume of IEEE Std. 1003.1-200x. The header  
 29061 for the next file entry in the archive shall be written directly after the last octet of the file entry  
 29062 preceding it. A header denoting the file name **TRAILER!!!** indicates the end of the archive; the  
 29063 contents of octets in the last block of the archive following such a header are undefined.

29064 **EXIT STATUS**

29065 The following exit values shall be returned:

29066     **0** All files were processed successfully.

29067     **>0** An error occurred.

29068 **CONSEQUENCES OF ERRORS**

29069 If *pax* cannot create a file or a link when reading an archive or cannot find a file when writing an  
 29070 archive, or cannot preserve the user ID, group ID, or file mode when the **-p** option is specified, a  
 29071 diagnostic message shall be written to standard error and a non-zero exit status shall be  
 29072 returned, but processing shall continue. In the case where *pax* cannot create a link to a file, *pax*  
 29073 shall not, by default, create a second copy of the file.

29074 If the extraction of a file from an archive is prematurely terminated by a signal or error, *pax* may  
 29075 have only partially extracted the file or (if the **-n** option was not specified) may have extracted a  
 29076 file of the same name as that specified by the user, but which is not the file the user wanted.  
 29077 Additionally, the file modes of extracted directories may have additional bits from the *S\_IRWXU*  
 29078 mask set as well as incorrect modification and access times.

29079 **APPLICATION USAGE**

29080 The **-p** (privileges) option was invented to reconcile differences between historical *tar* and *cpio*  
 29081 implementations. In particular, the two utilities use **-m** in diametrically opposed ways. The **-p**  
 29082 option also provides a consistent means of extending the ways in which future file attributes can  
 29083 be addressed, such as for enhanced security systems or high-performance files. Although it may  
 29084 seem complex, there are really two modes that are most commonly used:

29085     **-p e**     “Preserve everything”. This would be used by the historical superuser, someone with  
 29086 all the appropriate privileges, to preserve all aspects of the files as they are recorded in  
 29087 the archive. The **e** flag is the sum of **o** and **p**, and other implementation-dependent  
 29088 attributes.

29089     **-p p**     “Preserve” the file mode bits. This would be used by the user with regular privileges  
 29090 who wished to preserve aspects of the file other than the ownership. The file times are  
 29091 preserved by default, but two other flags are offered to disable these and use the time  
 29092 of extraction.

29093 The one path name per line format of standard input precludes path names containing  
 29094 <newline> characters. Although such path names violate the portable file name guidelines, they  
 29095 may exist and their presence may inhibit usage of *pax* within shell scripts. This problem is  
 29096 inherited from historical archive programs. The problem can be avoided by listing file name  
 29097 arguments on the command line instead of on standard input.

29098 It is almost certain that appropriate privileges are required for *pax* to accomplish parts of this  
 29099 volume of IEEE Std. 1003.1-200x. Specifically, creating files of type block special or character  
 29100 special, restoring file access times unless the files are owned by the user (the **-t** option), or  
 29101 preserving file owner, group, and mode (the **-p** option) all probably require appropriate  
 29102 privileges.

29103 In **read** mode, implementations are permitted to overwrite files when the archive has multiple  
 29104 members with the same name. This may fail if permissions on the first version of the file do not  
 29105 permit it to be overwritten.

29106 The *cpio* and **ustar** formats can only support files up to 8 gigabytes in size.

#### 29107 EXAMPLES

29108 The following command:

```
29109 pax -w -f /dev/rmt/lm .
```

29110 copies the contents of the current directory to tape drive 1, medium density (assuming historical  
 29111 System V device naming procedures. The historical BSD device name would be **/dev/rmt9**).

29112 The following commands:

```
29113 mkdir newdir
```

```
29114 pax -rw olddir newdir
```

29115 copy the *olddir* directory hierarchy to *newdir*.

```
29116 pax -r -s ',^//*usr//*,,' -f a.pax
```

29117 reads the archive **a.pax**, with all files rooted in **/usr** in the archive extracted relative to the current  
 29118 directory.

29119 Using the option:

```
29120 -o listopt="%M %(atime)T %(size)D %(name)s"
```

29121 overrides the default output description in Standard Output and instead writes:

```
29122 -rw-rw--- Jan 12 15:53 1492 /usr/foo/bar
```

29123 Using the options:

```
29124 -o listopt='%L\t%(size)D\n%.7' \
```

```
29125 -o listopt='(name)s\n%(ctime)T\n%T'
```

29126 overrides the default output description in Standard Output and instead writes:

```
29127 /usr/foo/bar -> /tmp 1492
```

```
29128 /usr/fo
```

```
29129 Jan 12 1991
```

```
29130 Jan 31 15:53
```

#### 29131 RATIONALE

29132 The *pax* utility was new, commissioned for the ISO POSIX-2:1993 standard. It represents a  
 29133 peaceful compromise between advocates of the historical *tar* and *cpio* utilities.

29134 A fundamental difference between *cpio* and *tar* was in the way directories were treated. The *cpio*  
 29135 utility did not treat directories differently from other files, and to select a directory and its  
 29136 contents required that each file in the hierarchy be explicitly specified. For *tar*, a directory  
 29137 matched every file in the file hierarchy it rooted.

29138 The *pax* utility offers both interfaces; by default, directories map into the file hierarchy they root.  
 29139 The **-d** option causes *pax* to skip any file not explicitly referenced, as *cpio* historically did. The *tar*  
 29140 **-style** behavior was chosen as the default because it was believed that this was the more  
 29141 common usage and because *tar* is the more commonly available interface, as it was historically  
 29142 provided on both System V and BSD implementations.

29143 The data interchange format specification in this volume of IEEE Std. 1003.1-200x requires that  
 29144 processes with “appropriate privileges” shall always restore the ownership and permissions of

29145 extracted files exactly as archived. If viewed from the historic equivalence between superuser  
29146 and “appropriate privileges”, there are two problems with this requirement. First, users running  
29147 as superusers may unknowingly set dangerous permissions on extracted files. Second, it is  
29148 needlessly limiting, in that superusers cannot extract files and own them as superuser unless the  
29149 archive was created by the superuser. (It should be noted that restoration of ownerships and  
29150 permissions for the superuser, by default, is historical practice in *cpio*, but not in *tar*.) In order to  
29151 avoid these two problems, the *pax* specification has an additional “privilege” mechanism, the **-p**  
29152 option. Only a *pax* invocation with the privileges needed, and which has the **-p** option set using  
29153 the **e** specification character, has the “appropriate privilege” to restore full ownership and  
29154 permission information.

29155 Note also that this volume of IEEE Std. 1003.1-200x requires that the file ownership and access  
29156 permissions shall be set, on extraction, in the same fashion as the *creat()* function when provided  
29157 the mode stored in the archive. This means that the file creation mask of the user is applied to  
29158 the file permissions.

29159 Users should note that directories may be created by *pax* while extracting files with permissions  
29160 that are different from those that existed at the time the archive was created. When extracting  
29161 sensitive information into a directory hierarchy that no longer exists, users are encouraged to set  
29162 their file creation mask appropriately to protect these files during extraction.

29163 The table of contents output is written to standard output to facilitate pipeline processing.

29164 The one path name per line format of standard input precludes path names containing  
29165 <newline> characters. Although such path names violate the portable file name guidelines, they  
29166 may exist and their presence may inhibit usage of *pax* within shell scripts. This problem is  
29167 inherited from historical archive programs. The problem can be avoided by listing file name  
29168 arguments on the command line instead of on standard input.

29169 An early proposal had hard links displaying for all path names. This was removed because it  
29170 complicates the output of the case where **-v** is not specified and does not match historical *cpio*  
29171 usage. The hard-link information is available in the **-v** display.

29172 The archive formats inherited from the POSIX.1-1990 standard have certain restrictions that  
29173 have been brought along from historical usage. For example, there are restrictions on the length  
29174 of path names stored in the archive. When *pax* is used in **copy( -rw)** mode (copying directory  
29175 hierarchies), the ability to use extensions from the **-xpax** format overcomes these restrictions.

29176 The default *blocksize* value of 5 120 bytes for *cpio* was selected because it is one of the standard  
29177 block-size values for *cpio*, set when the **-B** option is specified. (The other default block-size value  
29178 for *cpio* is 512 bytes, and this was considered to be too small.) The default block value of 10 240  
29179 bytes for *tar* was selected because that is the standard block-size value for BSD *tar*. The  
29180 maximum block size of 32 256 bytes ( $2^{15}$ –512 bytes) is the largest multiple of 512 bytes that fits  
29181 into a signed 16-bit tape controller transfer register. There are known limitations in some  
29182 historical systems that would prevent larger blocks from being accepted. Historical values were  
29183 chosen to improve compatibility with historical scripts using *dd* or similar utilities to manipulate  
29184 archives. Also, default block sizes for any file type other than character special file has been  
29185 deleted from this volume of IEEE Std. 1003.1-200x as unimportant and not likely to affect the  
29186 structure of the resulting archive.

29187 Implementations are permitted to modify the block-size value based on the archive format or  
29188 the device to which the archive is being written. This is to provide implementations with the  
29189 opportunity to take advantage of special types of devices, and it should not be used without a  
29190 great deal of consideration as it almost certainly decreases archive portability.

29191 The intended use of the **-n** option was to permit extraction of one or more files from the archive  
29192 without processing the entire archive. This was viewed by the standard developers as offering

significant performance advantages over historical implementations. The `-n` option in early proposals had three effects; the first was to cause special characters in patterns to not be treated specially. The second was to cause only the first file that matched a pattern to be extracted. The third was to cause *pax* to write a diagnostic message to standard error when no file was found matching a specified pattern. Only the second behavior is retained by this volume of IEEE Std. 1003.1-200x, for many reasons. First, it is in general not acceptable for a single option to have multiple effects. Second, the ability to make pattern matching characters act as normal characters is useful for parts of *pax* other than file extraction. Third, a finer degree of control over the special characters is useful because users may wish to normalize only a single special character in a single file name. Fourth, given a more general escape mechanism, the previous behavior of the `-n` option can be easily obtained using the `-s` option or a *sed* script. Finally, writing a diagnostic message when a pattern specified by the user is unmatched by any file is useful behavior in all cases.

In this version, the `-n` was removed from the **copy** mode synopsis of *pax*; it is inapplicable because there are no pattern operands specified in this mode.

There is another method than *pax* for copying subtrees in IEEE Std. 1003.1-200x described as part of the *cp* utility. Both methods are historical practice: *cp* provides a simpler, more intuitive interface, while *pax* offers a finer granularity of control. Each provides additional functionality to the other; in particular, *pax* maintains the hard-link structure of the hierarchy while *cp* does not. It is the intention of the standard developers that the results be similar (using appropriate option combinations in both utilities). The results are not required to be identical; there seemed insufficient gain to applications to balance the difficulty of implementations having to guarantee that the results would be exactly identical.

A single archive may span more than one file. It is suggested that implementations provide informative messages to the user on standard error whenever the archive file is changed.

The `-d` option (do not create intermediate directories not listed in the archive) found in early proposals was originally provided as a complement to the historic `-d` option of *cpio*. It has been deleted.

The `-s` option in early proposals specified a subset of the substitution command from the *ed* utility. As there was no reason for only a subset to be supported, the `-s` option is now compatible with the current *ed* specification. Since the delimiter can be any non-null character, the following usage with single spaces is valid:

```
pax -s " foo bar " ...
```

The `-t` option (specify an implementation-dependent identifier naming an input or output device) found in early proposals has been deleted because it is not historical practice and is of limited utility. In particular, historic versions of neither *cpio* nor *tar* had the concept of devices that were not mapped into the file system; if the devices are mapped into the file system, the `-f` option is sufficient.

The default behavior of *pax* with regard to file modification times is the same as historical implementations of *tar*. It is not the historical behavior of *cpio*.

Because the `-i` option uses `/dev/tty`, utilities without a controlling terminal are not able to use this option.

The `-y` option, found in early proposals, has been deleted because a line containing a single period for the `-i` option has equivalent functionality. The special lines for the `-i` option (a single period and the empty line) are historical practice in *cpio*.

In early drafts, an `-echarmap` option was included to increase portability of files between systems using different coded character sets. This option was omitted because it was apparent that

29240 consensus could not be formed for it. In this version, the use of UTF-8 should be an adequate  
29241 substitute.

29242 The **-k** option was added to address international concerns about the dangers involved in the  
29243 character set transformations of **-e** (if the target character set were different than the source, the  
29244 file names might be transformed into names matching existing files) and also was made more  
29245 general to protect files transferred between file systems with different {NAME\_MAX} values  
29246 (truncating a file name on a smaller system might also inadvertently overwrite existing files). As  
29247 stated, it prevents any overwriting, even if the target file is older than the source. This version  
29248 adds more granularity of options to solve this problem by introducing the **-oinvalid=** option—  
29249 specifically the UTF-8 action. (Note that an existing file that is named with a UTF-8 encoding is  
29250 still subject to overwriting in this case. The **-k** option closes that loophole.)

29251 Some of the file characteristics referenced in this volume of IEEE Std. 1003.1-200x might not be  
29252 supported by some archive formats. For example, neither the *tar* nor *cpio* formats contain the file  
29253 access time. For this reason, the **e** specification character has been provided, intended to cause all  
29254 file characteristics specified in the archive to be retained.

29255 It is required that extracted directories, by default, have their access and modification times and  
29256 permissions set to the values specified in the archive. This has obvious problems in that the  
29257 directories are almost certainly modified after being extracted and that directory permissions  
29258 may not permit file creation. One possible solution is to create directories with the mode  
29259 specified in the archive, as modified by the *umask* of the user, with sufficient permissions to  
29260 allow file creation. After all files have been extracted, *pax* would then reset the access and  
29261 modification times and permissions as necessary.

29262 The list-mode formatting description borrows heavily from the one defined by the *printf* utility.  
29263 However, since there is no separate operand list to get conversion arguments, the format was  
29264 extended to allow specifying the name of the conversion argument as part of the conversion  
29265 specification.

29266 The **T** specifier allows time fields to be displayed in any of the date formats. Unlike the *ls* utility,  
29267 *pax* does not adjust the format when the date is less than six months in the past. This makes  
29268 parsing the output more predictable.

29269 The **D** specifier handles the ability to display the major/minor or file size, as with *ls*, by using  
29270 **%-8(size)D**.

29271 The **L** specifier handles the *ls* display for symbolic links.

29272 Conversion specifiers were added to generate existing known types used for *ls*.

### 29273 **pax Interchange Format**

29274 The new POSIX data interchange format was developed primarily to satisfy international  
29275 concerns that the **ustar** and *cpio* formats did not provide for file, user, and group names encoded  
29276 in characters outside a subset of the ISO/IEC 646:1991 standard. The standard developers  
29277 realized that this new POSIX data interchange format should be very extensible because there  
29278 were other requirements they foresaw in the near future:

- 29279 • Support international character encodings and locale information
- 29280 • Support security information (ACLs, and so on)
- 29281 • Support future file types, such as realtime or contiguous files
- 29282 • Include data areas for implementation use



|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 29283 | <ul style="list-style-type: none"> <li>• Support systems with words larger than 32 bits and timers with subsecond granularity</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29284 | The following were not goals for this format because these are better handled by separate utilities or are inappropriate for a portable format:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29285 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29286 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29287 | <ul style="list-style-type: none"> <li>• Encryption</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29288 | <ul style="list-style-type: none"> <li>• Compression</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29289 | <ul style="list-style-type: none"> <li>• Data translation between locales and codesets</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 29290 | <ul style="list-style-type: none"> <li>• <i>inode</i> storage</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29291 | The format chosen to support the goals is an extension of the <b>ustar</b> format. Of the two formats previously available, only the <b>ustar</b> format was selected for extensions because:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 29292 | <ul style="list-style-type: none"> <li>• It was easier to extend in an upward-compatible way. It offered version flags and header block type fields with room for future standardization. The <i>cpio</i> format, while possessing a more flexible file naming methodology, could not be extended without breaking some theoretical implementation or using a dummy file name that could be a legitimate file name.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                          |
| 29293 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29294 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29295 | <ul style="list-style-type: none"> <li>• Industry experience since the original “tar wars” fought in developing the ISO POSIX-1 standard has clearly been in favor of the <b>ustar</b> format, which is generally the default output format selected for <i>pax</i> implementations on new systems.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29296 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29297 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29298 | The new format was designed with one additional goal in mind: reasonable behavior when an older <i>tar</i> or <i>pax</i> utility happened to read an archive. Since the POSIX.1-1990 standard mandated that a “format-reading utility” had to treat unrecognized <i>typeflag</i> values as regular files, this allowed the format to include all the extended information in a pseudo-regular file that preceded each real file. An option is given that allows the archive creator to set up reasonable names for these files on the older systems. Also, the normative text suggests that reasonable file access values be used for this <b>ustar</b> header block. Making these header files inaccessible for convenient reading and deleting would not be reasonable. File permissions of 600 or 700 are suggested. |
| 29299 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29300 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29301 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29302 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29303 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29304 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29305 | The <b>ustar</b> <i>typeflag</i> field was used to accommodate the additional functionality of the new format rather than magic or version because the POSIX.1-1990 standard (and, by reference, the previous version of <i>pax</i> ), mandated the behavior of the format-reading utility when it encountered an unknown <i>typeflag</i> , but was silent about the other two fields.                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 29306 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29307 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29308 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29309 | Early proposals of the first revision to IEEE Std. 1003.1-200x contained a proposed archive format that was based on compatibility with the standard for tape files (ISO 1001, similar to the format used historically on many mainframes and minicomputers). This format was overly complex and required considerable overhead in volume and header records. Furthermore, the standard developers felt that it would not be acceptable to the community of POSIX developers, so it was later changed to be a format more closely related to historical practice on POSIX systems.                                                                                                                                                                                                                                      |
| 29310 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29311 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29312 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29313 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29314 | The prefix and name split of path names in <b>ustar</b> was replaced by the single path extended header record for simplicity.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 29315 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29316 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29317 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29318 | The concept of a global extended header ( <i>typeflag</i> g) was controversial. If this were applied to an archive being recorded on magnetic tape, a few unreadable blocks at the beginning of the tape could be a serious problem; a utility attempting to extract as many files as possible from a damaged archive could lose a large percentage of file header information in this case. However, if the archive were on a reliable medium, such as a CD-ROM, the global extended header offers considerable potential size reductions by eliminating redundant information. Thus, the text warns against using the global method for unreliable media and provides a method for implanting global information in the extended header for each file, rather than in the <i>typeflag</i> g records.                  |
| 29319 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29320 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29321 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29322 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29323 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29324 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29325 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29326 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29327 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 29328 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |

29329 No facility for data translation or filtering on a per-file basis is included because the standard  
 29330 developers could not invent an interface that would allow this in an efficient manner. If a filter,  
 29331 such as encryption or compression, is to be applied to all the files, it is more efficient to apply the  
 29332 filter to the entire archive as a single file. The standard developers considered interfaces that  
 29333 would invoke a shell script for each file going into or out of the archive, but the system overhead  
 29334 in this approach was considered to be too high.

29335 One such approach would be to have **filter=** records that give a path name for an executable.  
 29336 When the program is invoked, the file and archive would be open for standard input/output  
 29337 and all the header fields would be available as environment variables or command-line  
 29338 arguments. The standard developers did discuss such schemes, but they were omitted from  
 29339 IEEE Std. 1003.1-200x due to concerns about excessive overhead. Also, the program itself would  
 29340 need to be in the archive if it were to be used portably.

29341 There is currently no portable means of identifying the character set(s) used for a file in the file  
 29342 system. Therefore, *pax* has not been given a mechanism to generate charset records  
 29343 automatically. The only portable means of doing this is for the user to write the archive using the  
 29344 **-ocharset=string** command line option. This assumes that all of the files in the archive use the  
 29345 same encoding. The “implementation-dependent” text is included to allow for a system that can  
 29346 identify the encodings used for each of its files.

29347 The table of standards that accompanies the charset record description is acknowledged to be  
 29348 very limited. Only a limited number of character set standards is reasonable for maximal  
 29349 interchange. Any character set is, of course, possible by prior agreement. It was suggested that  
 29350 EBCDIC be listed, but it was omitted because it is not defined by a formal standard. Formal  
 29351 standards, and then only those with reasonably large followings, can be included here, simply as  
 29352 a matter of practicality. The *<value>s* represent names of officially registered charactersets in the  
 29353 format required by the ISO 2375:1985 standard.

29354 The normal comma or *<blank>*-separated list rules are not followed in the case of keyword  
 29355 options to allow ease of argument parsing for *getopts*.

29356 Further information on character encodings is in **pax Archive Character Set Encoding/Decoding**  
 29357 on page 768.

29358 The standard developers have reserved keyword name space for vendor extensions. It is  
 29359 suggested that the format to be used is:

29360 *VENDOR.keyword*

29361 where *VENDOR* is the name of the vendor or organization in all uppercase letters. It is further  
 29362 suggested that the keyword following the period be named differently than any of the standard  
 29363 keywords so that it could be used for future standardization, if appropriate, by omitting the  
 29364 *VENDOR* prefix.

29365 The *<length>* field in the extended header record was included to make it simpler to step  
 29366 through the records, even if a record contains an unknown format (to a particular *pax*) with  
 29367 complex interactions of special characters. It also provides a minor integrity checkpoint within  
 29368 the records to aid a program attempting to recover files from a damaged archive.

29369 There are no extended header versions of the *devmajor* and *devminor* fields because the  
 29370 unspecified format **ustar** header field should be sufficient. If they are not, vendor-specific  
 29371 extended keywords (such as *VENDOR.devmajor*) should be used.

29372 Device and *i*-number labeling of files was not adopted from *cpio*; files are interchanged strictly  
 29373 on a symbolic name basis, as in **ustar**.

29374 Just as with the **ustar** format descriptions, the new format makes no special arrangements for  
 29375 multi-volume archives. Each of the *pax* archive types is assumed to be inside a single POSIX file  
 29376 and splitting that file over multiple volumes (diskettes, tape cartridges, and so on), processing  
 29377 their labels, and mounting each in the proper sequence are considered to be implementation  
 29378 details that cannot be described portably.

29379 The *pax* format is intended for interchange, not only for backup on a single (family of) systems. It  
 29380 is not as densely packed as might be possible for backup:

- 29381 • It contains information as coded characters that could be coded in binary.
- 29382 • It identifies extended records with name fields that could be omitted in favor of a fixed-field  
 29383 layout.
- 29384 • It translates names into a portable character set and identifies locale-related information,  
 29385 both of which are probably unnecessary for backup.

29386 The requirements on restoring from an archive are slightly different from the historical wording,  
 29387 allowing for non-monolithic privilege to bring forward as much as possible. In particular,  
 29388 attributes such as “high performance file” might be broadly but not universally granted while  
 29389 set-user-ID or *chown()* might be much more restricted. There is no implication in  
 29390 IEEE Std. 1003.1-200x that the security information be honored after it is restored to the file  
 29391 hierarchy, in spite of what might be improperly inferred by the silence on that topic. That is a  
 29392 topic for another standard.

29393 Links are recorded in the fashion described here because a link can be to any file type. It is  
 29394 desirable in general to be able to restore part of an archive selectively and restore all of those  
 29395 files completely. If the data is not associated with each link, it is not possible to do this.  
 29396 However, the data associated with a file can be large, and when selective restoration is not  
 29397 needed, this can be a significant burden. The archive is structured so that files that have no  
 29398 associated data can always be restored by the name of any link name of any link, and the user  
 29399 may choose whether data is recorded with each instance of a file that contains data. The format  
 29400 permits mixing of both types of links in a single archive; this can be done for special needs, and  
 29401 *pax* is expected to interpret such archives on input properly, despite the fact that there is no *pax*  
 29402 option that would force this mixed case on output. (When **-o linkdata** is used, the output must  
 29403 contain the duplicate data, but the implementation is free to include it or omit it when **-o**  
 29404 **linkdata** is not used.)

29405 The time values are included as extended header records for those implementations needing  
 29406 more than the eleven octal digits allowed by the **ustar** format. Even though some  
 29407 implementations can support finer file-time granularities than seconds, the normative text  
 29408 requires support only for seconds since the Epoch because the ISO POSIX-1 standard states them  
 29409 that way. The **ustar** format includes only *mtime*; the new format adds *atime* and *ctime* for  
 29410 symmetry. The *atime* access time restored to the file system will be affected by the **-p a** and **-p e**  
 29411 options. The *ctime* creation time (actually *inode* modification time) is described with  
 29412 “appropriate privilege” so that it can be ignored when writing to the file system. POSIX does not  
 29413 provide a portable means to change file creation time. Nothing is intended to prevent a non-  
 29414 portable implementation of *pax* from restoring the value.

29415 The *gid*, *size*, and *uid* extended header records were included to allow expansion beyond the  
 29416 sizes specified in the regular *tar* header. New file system architectures are emerging that will  
 29417 exhaust the 12-digit size field. There are probably not many systems requiring more than 8 digits  
 29418 for user and group IDs, but the extended header values were included for completeness,  
 29419 allowing overrides for all of the decimal values in the *tar* header.

29420 The standard developers intended to describe the effective results of *pax* with regard to file  
 29421 ownerships and permissions; implementations are not restricted in timing or sequencing the

29422 restoration of such, provided the results are as specified.

29423 Much of the text describing the extended headers refers to use in “**write** or **copy** modes”. The  
29424 **copy** mode references are due to the normative text: “The effect of the copy shall be as if the  
29425 copied files were written to an archive file and then subsequently extracted ...”. There is  
29426 certainly no way to test whether *pax* is actually generating the extended headers in **copy** mode,  
29427 but the effects must be as if it had.

#### 29428 **pax Archive Character Set Encoding/Decoding**

29429 There is a need to exchange archives of files between systems of different native codesets. File  
29430 names, group names, and user names must be preserved to the fullest extent possible when an  
29431 archive is read on the receiving platform. Translation of the contents of files is not within the  
29432 scope of the *pax* utility.

29433 There will also be the need to represent glyphs that are not available on the receiving platform.  
29434 (A *glyph* is commonly called a character, but without any reference to a specific encoding of that  
29435 character. The term *glyph* refers to the symbol itself.) These unsupported glyphs cannot be  
29436 automatically folded to the local set of glyphs due to the chance of collisions. This could result in  
29437 overwriting previous extracted files from the archive or pre-existing files on the system.

29438 For these reasons, the codeset used to represent glyphs within the extended header records of  
29439 the *pax* archive must be sufficiently rich to handle all commonly used character sets. The fields  
29440 requiring translation include, at a minimum, file names, user names, group names, and link path  
29441 names. Implementations may wish to have localized extended keywords that use non-portable  
29442 characters.

29443 The standard developers considered the following options:

- 29444 • The archive creator specifies the well-defined name of the source codeset. The receiver must  
29445 then recognize the codeset name and perform the appropriate translations to the destination  
29446 codeset.
- 29447 • The archive creator includes within the archive the character mapping table for the source  
29448 codeset used to encode extended header records. The receiver must then read the character  
29449 mapping table and perform the appropriate translations to the destination codeset.
- 29450 • The archive creator translates the extended header records in the source codeset into a  
29451 canonical form. The receiver must then perform the appropriate translations to the  
29452 destination codeset.

29453 The approach that incorporates the name of the source codeset poses the problem of codeset  
29454 name registration, and makes the archive useless to *pax* archive decoders that do not recognize  
29455 that codeset.

29456 Because parts of an archive may be corrupted, the standard developers felt that including the  
29457 character map of the source codeset was too fragile. The loss of this one key component could  
29458 result in making the entire archive useless. (The difference between this and the global extended  
29459 header decision was that the latter has a workaround—duplicating extended header records on  
29460 unreliable media—but this would be too burdensome for large character set maps.)

29461 Both of the above approaches also put an undue burden on the *pax* archive receiver to handle the  
29462 cross-product of all source and destination codesets.

29463 To simplify the translation from the source codeset to the canonical form and from the canonical  
29464 form to the destination codeset, the standard developers decided that the internal representation  
29465 should be a stateless encoding. A stateless encoding is one where each codepoint has the same  
29466 meaning, without regard to the decoder being in a specific state. An example of a stateful

29467 encoding would be the Japanese Shift-JIS; an example of a stateless encoding would be the  
29468 ISO/IEC 646: 1991 standard (equivalent to 7-bit ASCII).

29469 For these reasons, the standard developers decided to adopt a canonical format for the  
29470 representation of file information strings. The obvious, well-endorsed candidate is the  
29471 ISO/IEC 10646-1: 1993 standard (based in part on Unicode), which can be used to represent the  
29472 glyphs of virtually all standardized character sets. The standard developers initially agreed upon  
29473 using UCS2 (16-bit Unicode) as the internal representation. This repertoire of glyphs provides a  
29474 sufficiently rich set to represent all commonly-used codesets.

29475 However, the standard developers found that the 16-bit Unicode representation had some  
29476 problems. It forced the issue of standardizing byte ordering. The 2-byte length of each character  
29477 made the extended header records twice as long for the case of strings coded entirely from  
29478 historical 7-bit ASCII. For these reasons, the standard developers chose the UTF-8 defined in the  
29479 ISO/IEC 10646-1: 1993 standard. This multi-byte representation encodes UCS2 or UCS4  
29480 characters reliably and deterministically, eliminating the need for a canonical byte ordering. In  
29481 addition, NUL octets and other characters possibly confusing to POSIX file systems do not  
29482 appear, except to represent themselves. It was realized that certain national codesets take up  
29483 more space after the encoding, due to their placement within the UCS range; it was felt that the  
29484 usefulness of the encoding of the names outweighs the disadvantage of size increase for file,  
29485 user, and group names.

29486 The encoding of UTF-8 is as follows:

| 29487 | UCS4 Hex Encoding | UTF-8 Binary Encoding                                 |
|-------|-------------------|-------------------------------------------------------|
| 29488 | 00000000-0000007F | 0xxxxxxx                                              |
| 29489 | 00000080-000007FF | 110xxxxx 10xxxxxx                                     |
| 29490 | 00000800-0000FFFF | 1110xxxx 10xxxxxx 10xxxxxx                            |
| 29491 | 00010000-001FFFFF | 11110xxx 10xxxxxx 10xxxxxx 10xxxxxx                   |
| 29492 | 00200000-03FFFFFF | 111110xx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx          |
| 29493 | 04000000-7FFFFFFF | 1111110x 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx 10xxxxxx |

29494 where each 'x' represents a bit value from the character being translated.

## 29495 **ustar Interchange Format**

29496 The description of the **ustar** format reflects numerous enhancements over pre-1988 versions of  
29497 the historical *tar* utility. The goal of these changes was not only to provide the functional  
29498 enhancements desired, but also to retain compatibility between new and old versions. This  
29499 compatibility has been retained. Archives written using the old archive format are compatible  
29500 with the new format.

29501 Implementors should be aware that the previous file format did not include a mechanism to  
29502 archive directory type files. For this reason, the convention of using a file name ending with  
29503 slash was adopted to specify a directory on the archive.

29504 The total size of the *name* and *prefix* fields have been set to meet the minimum requirements for  
29505 {PATH\_MAX}. If a path name will fit within the *name* field, it is recommended that the path  
29506 name be stored there without the use of the *prefix* field. Although the name field is known to be  
29507 too small to contain {PATH\_MAX} characters, the value was not changed in this version of the  
29508 archive file format to retain backward compatibility, and instead the prefix was introduced.  
29509 Also, because of the earlier version of the format, there is no way to remove the restriction on the  
29510 *linkname* field being limited in size to just that of the *name* field.

29511 The *size* field is required to be meaningful in all implementation extensions, although it could be  
29512 zero. This is required so that the data blocks can always be properly counted.

29513 It is suggested that if device special files need to be represented that cannot be represented in the  
 29514 standard format that one of the extension types (A-Z) be used, and that the additional  
 29515 information for the special file be represented as data and be reflected in the *size* field.

29516 Attempting to restore a special file type, where it is converted to ordinary data and conflicts  
 29517 with an existing file name, need not be specially detected by the utility. If run as an ordinary  
 29518 user, *pax* should not be able to overwrite the entries in, for example, */dev* in any case (whether  
 29519 the file is converted to another type or not). If run as a privileged user, it should be able to do so,  
 29520 and it would be considered a bug if it did not. The same is true of ordinary data files and  
 29521 similarly named special files; it is impossible to anticipate the needs of the user (who could  
 29522 really intend to overwrite the file), so the behavior should be predictable (and thus regular) and  
 29523 rely on the protection system as required.

29524 The value 7 in the *typeflag* field is intended to define how contiguous files can be stored in a  
 29525 **ustar** archive. IEEE Std. 1003.1-200x does not require the contiguous file extension, but does  
 29526 define a standard way of archiving such files so that all conforming systems can interpret these  
 29527 file types in a meaningful and consistent manner. On a system that does not support extended  
 29528 file types, the *pax* utility should do the best it can with the file and go on to the next.

29529 The file protection modes are those conventionally used by the *ls* utility. This is extended  
 29530 beyond the usage in the ISO POSIX-2 standard to support the “shared text” or “sticky” bit. It is  
 29531 intended that the conformance document should not document anything beyond the existence  
 29532 of and support of such a mode. Further extensions are expected to these bits, particularly with  
 29533 overloading the set-user-ID and set-group-ID flags.

#### 29534 **cpio Interchange Format**

29535 The reference to appropriate privilege in the *cpio* format refers to an error on standard output;  
 29536 the **ustar** format does not make comparable statements.

29537 The model for this format was the historical System V *cpio-c* data interchange format. This  
 29538 model documents the portable version of the *cpio* format and not the binary version. It has the  
 29539 flexibility to transfer data of any type described within IEEE Std. 1003.1-200x, yet is extensible to  
 29540 transfer data types specific to extensions beyond IEEE Std. 1003.1-200x (for example, contiguous  
 29541 files). Because it describes existing practice, there is no question of maintaining upward  
 29542 compatibility.

#### 29543 **cpio Header**

29544 There has been some concern that the size of the *c\_ino* field of the header is too small to handle  
 29545 those systems that have very large *inode* numbers. However, the *c\_ino* field in the header is used  
 29546 strictly as a hard-link resolution mechanism for archives. It is not necessarily the same value as  
 29547 the *inode* number of the file in the location from which that file is extracted.

29548 The name *c\_magic* is based on historical usage.

#### 29549 **cpio File Name**

29550 For most historical implementations of the *cpio* utility, {PATH\_MAX} octets can be used to  
 29551 describe the path name without the addition of any other header fields (the NUL character  
 29552 would be included in this count). {PATH\_MAX} is the minimum value for path name size,  
 29553 documented as 256 bytes. However, an implementation may use *c\_namesize* to determine the  
 29554 exact length of the path name. With the current description of the **<cpio.h>** header, this path  
 29555 name size can be as large as a number that is described in six octal digits.

29556 Two values are documented under the *c\_mode* field values to provide for extensibility for known  
 29557 file types:

29558 **Notes to Reviewers**29559 *This section with side shading will not appear in the final copy. - Ed.*

29560 Note that the sockets extension below needs to be integrated, now that sockets have been  
 29561 merged

29562 **0110 000** Reserved for contiguous files. The implementation may treat the rest of the  
 29563 information for this archive like a regular file. If this file type is undefined, the  
 29564 implementation may create the file as a regular file.

29565 **0140 000** Reserved for sockets. If this type is undefined on the target system, the  
 29566 implementation may decide to ignore this file type and output a warning message.

29567 This provides for extensibility of the *cpio* format while allowing for the ability to read old  
 29568 archives. Files of an unknown type may be read as “regular files” on some implementations. On  
 29569 a system that does not support extended file types, the *pax* utility should do the best it can with  
 29570 the file and go on to the next.

29571 **FUTURE DIRECTIONS**

29572 None.

29573 **SEE ALSO**

29574 *cp*, *ed*, *getopts*, *printf*, the System Interface Definitions volume of IEEE Std. 1003.1-200x, *<cpio.h>*,  
 29575 the System Interfaces volume of IEEE Std. 1003.1-200x, *chown()*, *creat()*, *mkdir()*, *stat()*, *write()*

29576 **CHANGE HISTORY**

29577 First released in Issue 4.

29578 **Issue 5**

29579 A note is added to the APPLICATION USAGE indicating that the *cpio* and *tar* formats can only  
 29580 support files up to 8 gigabytes in size.

29581 **Issue 6**29582 The *pax* utility is aligned with the IEEE P1003.2b draft standard:

- 29583 • Support has been added for symbolic links in the options and interchange formats.
- 29584 • A new format has been devised, based on extensions to *ustar*.
- 29585 • References to the “extended” *tar* and *cpio* formats derived from the POSIX.1-1990 standard
- 29586 have been changed to remove the “extended” adjective because this could cause confusion
- 29587 with the extended *tar* header added in this revision. (All references to *tar* are actually to
- 29588 **ustar**).

## 29589 NAME

29590 pr — print files

## 29591 SYNOPSIS

```
29592 pr [+page][-column][-adFmrt][-e[char][gap]][-h header][-i[char][gap]]
29593 xsi [-l lines][-n[char][width]][-o offset][-s[char]][-w width][-fp]
29594 [file...]
```

## 29595 DESCRIPTION

29596 The *pr* utility is a printing and pagination filter. If multiple input files are specified, each shall be  
 29597 read, formatted, and written to standard output. By default, the input shall be separated into 66-  
 29598 line pages, each with:

- 29599 • A 5-line header that includes the page number, date, time, and the path name of the file
- 29600 • A 5-line trailer consisting of blank lines

29601 If standard output is associated with a terminal, diagnostic messages shall be deferred until the  
 29602 *pr* utility has completed processing.

29603 When options specifying multi-column output are specified, output text columns shall be of  
 29604 equal width; input lines that do not fit into a text column shall be truncated. By default, text  
 29605 columns shall be separated with at least one <blank> character.

## 29606 OPTIONS

29607 The *pr* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 29608 Section 12.2, Utility Syntax Guidelines, except that: the *page* option has a '+' delimiter; *page* and  
 29609 *column* can be multi-digit numbers; some of the option-arguments are optional; and some of the  
 29610 option-arguments cannot be specified as separate arguments from the preceding option letter. In  
 29611 particular, the *-s* option does not allow the option letter to be separated from its argument, and  
 29612 the options *-e*, *-i*, and *-n* require that both arguments, if present, not be separated from the  
 29613 option letter.

29614 The following options shall be supported. In the following option descriptions, *column*, *lines*,  
 29615 *offset*, *page*, and *width* are positive decimal integers; *gap* is a non-negative decimal integer.

29616 *+page* Begin output at page number *page* of the formatted input.

29617 *-column* Produce multi-column output that is arranged in *column* columns (the default shall  
 29618 be 1) and is written down each column in the order in which the text is received  
 29619 from the input file. This option should not be used with *-m*. The options *-e* and *-i*  
 29620 shall be assumed for multiple text-column output. Whether or not text columns  
 29621 are produced with identical vertical lengths is unspecified, but a text column shall  
 29622 never exceed the length of the page (see the *-l* option). When used with *-t*, use the  
 29623 minimum number of lines to write the output.

29624 *-a* Modify the effect of the *-column* option so that the columns are filled across the  
 29625 page in a round-robin order (for example, when *column* is 2, the first input line  
 29626 heads column 1, the second heads column 2, the third is the second line in column  
 29627 1, and so on).

29628 *-d* Produce output that is double-spaced; append an extra <newline> character  
 29629 following every <newline> character found in the input.

29630 *-e[*char*][*gap*]*

29631 Expand each input <tab> character to the next greater column position specified  
 29632 by the formula  $n * \text{gap} + 1$ , where *n* is an integer > 0. If *gap* is zero or is omitted, it  
 29633 shall default to 8. All <tab> characters in the input shall be expanded into the  
 29634 appropriate number of <space> characters. If any non-digit character, *char*, is



|           |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----------|------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 29635     |                        | specified, it shall be used as the input <tab> character.                                                                                                                                                                                                                                                                                                                                                                      |
| 29636 XSI | <b>-f</b>              | Use a <form-feed> character for new pages, instead of the default behavior that uses a sequence of <newline> characters. Pause before beginning the first page if the standard output is associated with a terminal.                                                                                                                                                                                                           |
| 29637     |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29638     |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29639     | <b>-F</b>              | Use a <form-feed> character for new pages, instead of the default behavior that uses a sequence of <newline> characters.                                                                                                                                                                                                                                                                                                       |
| 29640     |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29641     | <b>-h header</b>       | Use the string <i>header</i> to replace the contents of the <i>file</i> operand in the page header.                                                                                                                                                                                                                                                                                                                            |
| 29642     | <b>-i[char][gap]</b>   |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29643     |                        | In output, replace multiple <space> characters with <tab> characters wherever two or more adjacent <space> characters reach column positions <i>gap</i> +1, 2* <i>gap</i> +1, 3* <i>gap</i> +1, and so on. If <i>gap</i> is zero or is omitted, default tab settings at every eighth column position shall be assumed. If any non-digit character, <i>char</i> , is specified, it shall be used as the output <tab> character. |
| 29644     |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29645     |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29646     |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29647     |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29648     | <b>-l lines</b>        | Override the 66-line default and reset the page length to <i>lines</i> . If <i>lines</i> is not greater than the sum of both the header and trailer depths (in lines), the <i>pr</i> utility shall suppress both the header and trailer, as if the <b>-t</b> option were in effect.                                                                                                                                            |
| 29649     |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29650     |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29651     | <b>-m</b>              | Merge files. Standard output shall be formatted so the <i>pr</i> utility writes one line from each file specified by a <i>file</i> operand, side by side into text columns of equal fixed widths, in terms of the number of column positions. Implementations shall support merging of at least nine <i>file</i> operands.                                                                                                     |
| 29652     |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29653     |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29654     |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29655     | <b>-n[char][width]</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29656     |                        | Provide <i>width</i> -digit line numbering (default for <i>width</i> shall be 5). The number shall occupy the first <i>width</i> column positions of each text column of default output or each line of <b>-m</b> output. If <i>char</i> (any non-digit character) is given, it shall be appended to the line number to separate it from whatever follows (default for <i>char</i> is a <tab> character).                      |
| 29657     |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29658     |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29659     |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29660     |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29661     | <b>-o offset</b>       | Each line of output shall be preceded by offset <space> <i>s</i> . If the <b>-o</b> option is not specified, the default offset shall be zero. The space taken is in addition to the output line width (see the <b>-w</b> option below).                                                                                                                                                                                       |
| 29662     |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29663     |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29664     | <b>-p</b>              | Pause before beginning each page if the standard output is directed to a terminal ( <i>pr</i> shall write an <alert> character to standard error and wait for a <carriage-return> character to be read on <i>/dev/tty</i> ).                                                                                                                                                                                                   |
| 29665     |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29666     |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29667     | <b>-r</b>              | Write no diagnostic reports on failure to open files.                                                                                                                                                                                                                                                                                                                                                                          |
| 29668     | <b>-s[char]</b>        | Separate text columns by the single character <i>char</i> instead of by the appropriate number of <space> characters (default for <i>char</i> shall be the <tab> character).                                                                                                                                                                                                                                                   |
| 29669     |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29670     | <b>-t</b>              | Write neither the five-line identifying header nor the five-line trailer usually supplied for each page. Quit writing after the last line of each file without spacing to the end of the page.                                                                                                                                                                                                                                 |
| 29671     |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29672     |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29673     | <b>-w width</b>        | Set the width of the line to <i>width</i> column positions for multiple text-column output only. If the <b>-w</b> option is not specified and the <b>-s</b> option is not specified, the default width shall be 72. If the <b>-w</b> option is not specified and the <b>-s</b> option is specified, the default width shall be 512.                                                                                            |
| 29674     |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29675     |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29676     |                        |                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 29677     |                        | For single column output, input lines shall not be truncated.                                                                                                                                                                                                                                                                                                                                                                  |

29678 **OPERANDS**

29679 The following operand shall be supported:

29680 *file* A path name of a file to be written. If no *file* operands are specified, or if a *file*  
 29681 operand is '-', the standard input shall be used.

29682 **STDIN**

29683 The standard input shall be used only if no *file* operands are specified, or if a *file* operand is '-'.  
 29684 See the INPUT FILES section.

29685 **INPUT FILES**

29686 The input files shall be text files.

29687 The file */dev/tty* is used to read responses required by the **-p** option.

29688 **ENVIRONMENT VARIABLES**

29689 The following environment variables shall affect the execution of *pr*:

29690 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 29691 If *LANG* is unset or null, the corresponding value from the implementation-  
 29692 dependent default locale shall be used. If any of the internationalization variables  
 29693 contains an invalid setting, the utility shall behave as if none of the variables had  
 29694 been defined.

29695 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 29696 internationalization variables.

29697 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 29698 characters (for example, single-byte as opposed to multi-byte characters in  
 29699 arguments and input files) and which characters are defined as printable (character  
 29700 class **print**). Non-printable characters are still written to standard output, but are  
 29701 not counted for the purpose for column-width and line-length calculations.

29702 *LC\_MESSAGES*

29703 Determine the locale that should be used to affect the format and contents of  
 29704 diagnostic messages written to standard error.

29705 *LC\_TIME* Determine the format of the date and time for use in writing header lines.

29706 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

29707 *TZ* Determine the timezone for use in writing header lines.

29708 **ASYNCHRONOUS EVENTS**

29709 If *pr* receives an interrupt while writing to a terminal, it shall flush all accumulated error  
 29710 messages to the screen before terminating.

29711 **STDOUT**

29712 The *pr* utility output shall be a paginated version of the original file (or files). This pagination  
 29713 shall be accomplished using either <form-feed> characters or a sequence of <newline>  
 29714 XSI characters, as controlled by the **-F** or **-f** option. Page headers shall be generated unless the **-t**  
 29715 option is specified. The page headers shall be of the form:

29716 "\n\n%s %s Page %d\n\n\n", <output of date>, <file>, <page number>

29717 In the POSIX locale, the <output of date> field, representing the date and time of last modification  
 29718 of the input file (or the current date and time if the input file is standard input), shall be  
 29719 equivalent to the output of the following command as it would appear if executed at the given  
 29720 time:

29721 date "+%b %e %H:%M %Y"

29722 without the trailing <newline> character, if the page being written is from standard input. If the  
 29723 page being written is not from standard input, in the POSIX locale, the same format shall be  
 29724 used, but the time used shall be the modification time of the file corresponding to *file* instead of  
 29725 the current time. When the *LC\_TIME* locale category is not set to the POSIX locale, a different  
 29726 format and order of presentation of this field may be used.

29727 If the standard input is used instead of a *file* operand, the <*file*> field shall be replaced by a null  
 29728 string.

29729 If the **-h** option is specified, the <*file*> field shall be replaced by the *header* argument.

## 29730 STDERR

29731 MAN Used for diagnostic messages and for alerting the terminal when **-p** is specified.

## 29732 OUTPUT FILES

29733 None.

## 29734 EXTENDED DESCRIPTION

29735 None.

## 29736 EXIT STATUS

29737 The following exit values shall be returned:

29738 0 Successful completion.

29739 >0 An error occurred.

## 29740 CONSEQUENCES OF ERRORS

29741 Default.

## 29742 APPLICATION USAGE

29743 None.

## 29744 EXAMPLES

29745 1. Print a numbered list of all files in the current directory:

29746 `ls -a | pr -n -h "Files in $(pwd)."`

29747 2. Print **file1** and **file2** as a double-spaced, three-column listing headed by “file list”:

29748 `pr -3d -h "file list" file1 file2`

29749 3. Write **file1** on **file2**, expanding tabs to columns 10, 19, 28, ...:

29750 `pr -e9 -t <file1 >file2`

## 29751 RATIONALE

29752 This utility is one of those that does not follow the Utility Syntax Guidelines because of its  
 29753 historical origins. The standard developers could have added new options that obeyed the  
 29754 guidelines (and marked the old options *obsolescent*) or devised an entirely new utility; there are  
 29755 examples of both actions in this volume of IEEE Std. 1003.1-200x. Because of its widespread use  
 29756 by historical applications, the standard developers decided to exempt this version of *pr* from  
 29757 many of the guidelines.

29758 Implementations are required to accept option-arguments to the **-h**, **-l**, **-o**, and **-w** options  
 29759 whether presented as part of the same argument or as a separate argument to *pr*, as suggested by  
 29760 the Utility Syntax Guidelines. The **-n** and **-s** options, however, are specified as in historical  
 29761 practice because they are frequently specified without their optional arguments. If a <blank>  
 29762 were allowed before the option-argument in these cases, a file operand could mistakenly be

- 29763 interpreted as an option-argument in historical applications.
- 29764 The text about the minimum number of lines in multi-column output was included to ensure  
 29765 that a best effort is made in balancing the length of the columns. There are known historical  
 29766 implementations in which, for example, 60-line files are listed by *pr -2* as one column of 56 lines  
 29767 and a second of 4. Although this is not a problem when a full page with headers and trailers is  
 29768 produced, it would be relatively useless when used with *-t*.
- 29769 Historical implementations of the *pr* utility have differed in the action taken for the *-f* option.  
 29770 BSD uses it as described here for the *-F* option; System V uses it to change trailing <newline>s  
 29771 on each page to a <form-feed> and, if standard output is a TTY device, sends an <alert> to  
 29772 standard error and reads a line from */dev/tty* before the first page. There were strong arguments  
 29773 from both sides of this issue concerning historical practice and additional arguments against the  
 29774 System V *-f* behavior, on the grounds that having the behavior of an option change depending  
 29775 on where output is directed was not a modular design. Therefore, the *-f* option is not specified  
 29776 and the *-F* option has been added.
- 29777 The <output of date> field in the *-l* format is specified only for the POSIX locale. As noted, the  
 29778 format can be different in other locales. No mechanism for defining this is present in this volume  
 29779 of IEEE Std. 1003.1-200x, as the appropriate vehicle is a messaging system; that is, the format  
 29780 should be specified as a “message”.
- 29781 **FUTURE DIRECTIONS**
- 29782 It is possible that a new interface that conforms to the Utility Syntax Guidelines will be  
 29783 introduced.
- 29784 **SEE ALSO**
- 29785 *expand, lp*
- 29786 **CHANGE HISTORY**
- 29787 First released in Issue 2.
- 29788 **Issue 4**
- 29789 Aligned with the ISO/IEC 9945-2: 1993 standard.
- 29790 **Issue 6**
- 29791 The following new requirements on POSIX implementations derive from alignment with the  
 29792 Single UNIX Specification:
- 29793 • The *-p* option is added.
- 29794 The normative text is reworded to avoid use of the term “must” for application requirements.

29795 **NAME**

29796       printf — write formatted output

29797 **SYNOPSIS**29798       printf *format*[*argument...*]29799 **DESCRIPTION**

29800       The *printf* utility shall write formatted operands to the standard output. The *argument* operands  
 29801       shall be formatted under control of the *format* operand.

29802 **OPTIONS**

29803       None.

29804 **OPERANDS**

29805       The following operands shall be supported:

29806       *format*       A string describing the format to use to write the remaining operands. See the  
 29807       EXTENDED DESCRIPTION section.

29808       *argument*    The strings to be written to standard output, under the control of *format*. See the  
 29809       EXTENDED DESCRIPTION section.

29810 **STDIN**

29811       Not used.

29812 **INPUT FILES**

29813       None.

29814 **ENVIRONMENT VARIABLES**29815       The following environment variables shall affect the execution of *printf*:

29816       *LANG*        Provide a default value for the internationalization variables that are unset or null.  
 29817       If *LANG* is unset or null, the corresponding value from the implementation-  
 29818       dependent default locale shall be used. If any of the internationalization variables  
 29819       contains an invalid setting, the utility shall behave as if none of the variables had  
 29820       been defined.

29821       *LC\_ALL*       If set to a non-empty string value, override the values of all the other  
 29822       internationalization variables.

29823       *LC\_CTYPE*    Determine the locale for the interpretation of sequences of bytes of text data as  
 29824       characters (for example, single-byte as opposed to multi-byte characters in  
 29825       arguments).

29826       *LC\_MESSAGES*  
 29827       Determine the locale that should be used to affect the format and contents of  
 29828       diagnostic messages written to standard error.

29829       *LC\_NUMERIC*  
 29830       Determine the locale for numeric formatting. It shall affect the format of numbers  
 29831       written using the *e*, *E*, *f*, *g*, and *G* conversion characters (if supported).

29832 **XSI**       *NLSPATH*   Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

29833 **ASYNCHRONOUS EVENTS**

29834       Default.

29835 **STDOUT**

29836 See the EXTENDED DESCRIPTION section.

29837 **STDERR**

29838 Used only for diagnostic messages.

29839 **OUTPUT FILES**

29840 None.

29841 **EXTENDED DESCRIPTION**

29842 The *format* operand shall be used as the *format* string described in the System Interface  
 29843 Definitions volume of IEEE Std. 1003.1-200x, Chapter 5, File Format Notation with the following  
 29844 exceptions:

- 29845 1. A <space> character in the format string, in any context other than a flag of a conversion  
 29846 specification, shall be treated as an ordinary character that is copied to the output.
- 29847 2. A 'Δ' character in the format string shall be treated as a 'Δ' character, not as a <space>  
 29848 character.
- 29849 3. In addition to the escape sequences shown in the System Interface Definitions volume of  
 29850 IEEE Std. 1003.1-200x, Chapter 5, File Format Notation ('\\', '\a', '\b', '\f', '\n',  
 29851 '\r', '\t', '\v'), "\ddd", where *ddd* is a one, two, or three-digit octal number, shall be  
 29852 written as a byte with the numeric value specified by the octal number.
- 29853 4. The implementation shall not precede or follow output from the *d* or *u* conversion  
 29854 specifications with <blank> characters not specified by the *format* operand.
- 29855 5. The implementation shall not precede output from the *o* conversion specification with  
 29856 zeros not specified by the *format* operand.
- 29857 6. The *e*, *E*, *f*, *g*, and *G* conversion specifications need not be supported.
- 29858 7. An additional conversion character, *b*, shall be supported as follows. The argument shall  
 29859 be taken to be a string that may contain backslash-escape sequences. The following  
 29860 backslash-escape sequences shall be supported:
  - 29861 — The escape sequences listed in the System Interface Definitions volume of  
 29862 IEEE Std. 1003.1-200x, Chapter 5, File Format Notation ('\\', '\a', '\b', '\f',  
 29863 '\n', '\r', '\t', '\v'), which shall be converted to the characters they represent
  - 29864 — "\ddd", where *ddd* is a zero, one, two, or three-digit octal number that shall be  
 29865 converted to a byte with the numeric value specified by the octal number
  - 29866 — '\c', which shall not be written and shall cause *printf* to ignore any remaining  
 29867 characters in the string operand containing it, any remaining string operands, and any  
 29868 additional characters in the *format* operand
- 29869 The interpretation of a backslash followed by any other sequence of characters is  
 29870 unspecified.
- 29871 Bytes from the converted string shall be written until the end of the string or the number of  
 29872 bytes indicated by the precision specification is reached. If the precision is omitted, it shall  
 29873 be taken to be infinite, so all bytes up to the end of the converted string shall be written.
- 29874 8. For each specification that consumes an argument, the next argument operand shall be  
 29875 evaluated and converted to the appropriate type for the conversion as specified below.
- 29876 9. The *format* operand shall be reused as often as necessary to satisfy the argument operands.  
 29877 Any extra *c* or *s* conversion specifications shall be evaluated as if a null string argument  
 29878 were supplied; other extra conversion specifications shall be evaluated as if a zero

29879 argument were supplied. If the *format* operand contains no conversion specifications and  
 29880 *argument* operands are present, the results are unspecified.

29881 10. If a character sequence in the *format* operand begins with a '%' character, but does not  
 29882 form a valid conversion specification, the behavior is unspecified.

29883 The *argument* operands shall be treated as strings if the corresponding conversion character is *b*,  
 29884 *c*, or *s*; otherwise, it shall be evaluated as a C constant, as described by the ISO C standard, with  
 29885 the following extensions:

- 29886 • A leading plus or minus sign shall be allowed.
- 29887 • If the leading character is a single-quote or double-quote, the value shall be the numeric  
 29888 value in the underlying codeset of the character following the single-quote or double-quote.

29889 If an argument operand cannot be completely converted into an internal value appropriate to  
 29890 the corresponding conversion specification, a diagnostic message shall be written to standard  
 29891 error and the utility shall not exit with a zero exit status, but shall continue processing any  
 29892 remaining operands and shall write the value accumulated at the time the error was detected to  
 29893 standard output.

#### 29894 EXIT STATUS

29895 The following exit values shall be returned:

29896 0 Successful completion.

29897 >0 An error occurred.

#### 29898 CONSEQUENCES OF ERRORS

29899 Default.

#### 29900 APPLICATION USAGE

29901 The floating-point formatting conversion specifications of *printf()* are not required because all  
 29902 arithmetic in the shell is integer arithmetic. The *awk* utility performs floating-point calculations  
 29903 and provides its own **printf** function. The *bc* utility can perform arbitrary-precision floating-  
 29904 point arithmetic, but does not provide extensive formatting capabilities. (This *printf* utility  
 29905 cannot really be used to format *bc* output; it does not support arbitrary precision.)  
 29906 Implementations are encouraged to support the floating-point conversions as an extension.

29907 Note that this *printf* utility, like the *printf()* function defined in the System Interfaces volume of  
 29908 IEEE Std. 1003.1-200x on which it is based, makes no special provision for dealing with multi-  
 29909 byte characters when using the *%c* conversion specification or when a precision is specified in a  
 29910 *%b* or *%s* conversion specification. Applications should be extremely cautious using either of  
 29911 these features when there are multi-byte characters in the character set.

29912 No provision is made in this volume of IEEE Std. 1003.1-200x which allows field widths and  
 29913 precisions to be specified as '\*' since the '\*' can be replaced directly in the *format* operand  
 29914 using shell variable substitution. Implementations can also provide this feature as an extension  
 29915 if they so choose.

29916 Hexadecimal character constants as defined in the ISO C standard are not recognized in the  
 29917 *format* operand because there is no consistent way to detect the end of the constant. Octal  
 29918 character constants are limited to, at most, three octal digits, but hexadecimal character  
 29919 constants are only terminated by a non-hex-digit character. In the ISO C standard, the "##"  
 29920 concatenation operator can be used to terminate a constant and follow it with a hexadecimal  
 29921 character to be written. In the shell, concatenation occurs before the *printf* utility has a chance to  
 29922 parse the end of the hexadecimal constant.

29923 The `%b` conversion specification is not part of the ISO C standard; it has been added here as a  
 29924 portable way to process backslash escapes expanded in string operands as provided by the *echo*  
 29925 utility. See also the APPLICATION USAGE section of *echo* on page 366 for ways to use *printf* as a  
 29926 replacement for all of the traditional versions of the *echo* utility.

29927 If an argument cannot be parsed correctly for the corresponding conversion specification, the  
 29928 *printf* utility is required to report an error. Thus, overflow and extraneous characters at the end  
 29929 of an argument being used for a numeric conversion shall be reported as errors.

## 29930 **Notes to Reviewers**

29931 *This section with side shading will not appear in the final copy. - Ed.*

29932 Is the following text normative? If so, where should it be moved to?

29933 It is not considered an error if an argument operand is not completely used for a `c` or `s`  
 29934 conversion or if a string operand's first or second character is used to get the numeric value of a  
 29935 character.

## 29936 **EXAMPLES**

29937 To alert the user and then print and read a series of prompts:

```
29938 printf "\aPlease fill in the following: \nName: "
29939 read name
29940 printf "Phone number: "
29941 read phone
```

29942 To read out a list of right and wrong answers from a file, calculate the percentage correctly, and  
 29943 print them out. The numbers are right-justified and separated by a single <tab> character. The  
 29944 percentage is written to one decimal place of accuracy:

```
29945 while read right wrong ; do
29946 percent=$(echo "scale=1;($right*100)/($right+$wrong)" | bc)
29947 printf "%2d right\t%2d wrong\t(%%s%%)\n" \
29948 $right $wrong $percent
29949 done < database_file
```

29950 The command:

```
29951 printf "%5d%4d\n" 1 21 321 4321 54321
```

29952 produces:

```
29953 1 21
29954 3214321
29955 54321 0
```

29956 Note that the *format* operand is used three times to print all of the given strings and that a `'0'`  
 29957 was supplied by *printf* to satisfy the last `%4d` conversion specification.

29958 The *printf* utility is required to notify the user when conversion errors are detected while  
 29959 producing numeric output; thus, the following results would be expected on an implementation  
 29960 with 32-bit twos-complement integers when `%d` is specified as the *format* operand:



29961  
29962  
29963  
29964  
29965  
29966  
29967

| Argument    | Standard Output | Diagnostic Output                         |
|-------------|-----------------|-------------------------------------------|
| 5a          | 5               | printf: "5a" not completely converted     |
| 9999999999  | 2147483647      | printf: "9999999999" arithmetic overflow  |
| -9999999999 | -2147483648     | printf: "-9999999999" arithmetic overflow |
| ABC         | 0               | printf: "ABC" expected numeric value      |

29968  
29969  
29970  
29971  
29972  
29973

The diagnostic message format is not specified, but these examples convey the type of information that should be reported. Note that the value shown on standard output is what would be expected as the return value from the *strtol()* function as defined in the System Interfaces volume of IEEE Std. 1003.1-200x. A similar correspondence exists between *%u* and *strtoul()* and *%e*, *%f*, and *%g* (if the implementation supports floating-point conversions) and *strtod()*.

29974

In a locale using the ISO/IEC 646: 1991 standard as the underlying codeset, the command:

29975

```
printf "%d\n" 3 +3 -3 \'3 \"+3 "'-3"
```

29976

produces:

29977

3    Numeric value of constant 3

29978

3    Numeric value of constant 3

29979

-3   Numeric value of constant -3

29980

51   Numeric value of the character '3' in the ISO/IEC 646: 1991 standard codeset

29981

43   Numeric value of the character '+' in the ISO/IEC 646: 1991 standard codeset

29982

45   Numeric value of the character '-' in the ISO/IEC 646: 1991 standard codeset

29983

Note that in a locale with multi-byte characters, the value of a character is intended to be the value of the equivalent of the **wchar\_t** representation of the character as described in the System Interfaces volume of IEEE Std. 1003.1-200x.

29984

29985

#### 29986 RATIONALE

29987

The *printf* utility was added to provide functionality that has historically been provided by *echo*.

29988

However, due to irreconcilable differences in the various versions of *echo* extant, the version has few special features, leaving those to this new *printf* utility, which is based on one in the Ninth Edition system.

29989

29990

29991

The EXTENDED DESCRIPTION section almost exactly matches the *printf()* function in the ISO C standard, although it is described in terms of the file format notation in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 5, File Format Notation.

29992

29993

#### 29994 FUTURE DIRECTIONS

29995

None.

#### 29996 SEE ALSO

29997

*awk*, *bc*, *echo*, the System Interfaces volume of IEEE Std. 1003.1-200x, *printf()*

#### 29998 CHANGE HISTORY

29999

First released in Issue 4.

## 30000 NAME

30001        *prs* — print an SCCS file (**DEVELOPMENT**)

## 30002 SYNOPSIS

30003 XSI        *prs* [-a][-d *dataspec*][-r[*SID*]] *file...*

30004 XSI        *prs* [ -e | -l ] -c *cutoff* [-d *dataspec*] *file...*

30005 XSI        *prs* [ -e | -l ] -r[*SID*][-d *dataspec*]*file...*

30006

## 30007 DESCRIPTION

30008        The *prs* utility shall write to standard output parts or all of an SCCS file in a user-supplied  
30009        format.

## 30010 OPTIONS

30011        The *prs* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
30012        Section 12.2, Utility Syntax Guidelines, except that the **-r** option has an optional option-  
30013        argument. This optional option-argument cannot be presented as a separate argument. The  
30014        following options shall be supported:

30015        **-d *dataspec***    Specify the output data specification. The *dataspec* is a string consisting of SCCS  
30016        file *data keywords* (see **Data Keywords** on page 783) interspersed with optional  
30017        user-supplied text.

30018        **-r[*SID*]**        Specify the SCCS identification string (*SID*) of a delta for which information is  
30019        desired. If no *SID* option-argument is specified, the *SID* of the most recently  
30020        created delta is assumed.

30021        **-e**                Request information for all deltas created earlier than and including the delta  
30022        designated via the **-r** option or the date-time given by the **-c** option.

30023        **-l**                Request information for all deltas created later than and including the delta  
30024        designated via the **-r** option or the date-time given by the **-c** option.

30025        **-c *cutoff***        Indicate the *cutoff* date-time, in the form:

30026                *YY*[*MM* [*DD* [*HH* [*MM* [*SS*] ] ] ] ]

30027                For the *YY* component, values in the range [69-99] shall refer to years in the  
30028                twentieth century (1969 to 1999 inclusive); values in the range [00-68] shall refer to  
30029                years in the twenty-first century (2000 to 2068 inclusive).

30030                No changes (deltas) to the SCCS file that were created after the specified *cutoff*  
30031                date-time shall be included in the output. Units omitted from the date-time default  
30032                to their maximum possible values; for example, **-c 7502** is equivalent to  
30033                **-c 750228235959**.

30034        **-a**                Request writing of information for both removed, that is, *delta type*=*R* (see *rm del* on  
30035        page 865) and existing, that is, *delta type*=*D*, deltas. If the **-a** option is not specified,  
30036        information for existing deltas only shall be provided.

## 30037 OPERANDS

30038        The following operand shall be supported:

30039        *file*            A path name of an existing SCCS file or a directory. If *file* is a directory, *prs* behaves  
30040        as though each file in the directory were specified as a named file, except that  
30041        non-SCCS files (last component of the path name does not begin with **s**.) and  
30042        unreadable files are silently ignored.

30043 If a single instance *file* is specified as *'-'*, the standard input shall be read; each  
 30044 line of the standard input shall be taken to be the name of an SCCS file to be  
 30045 processed. Non-SCCS files and unreadable files are silently ignored.

#### 30046 **STDIN**

30047 The standard input shall be a text file used only when the *file* operand is specified as *'-'*. Each  
 30048 line of the text file shall be interpreted as an SCCS path name.

#### 30049 **INPUT FILES**

30050 Any SCCS files displayed are files of an unspecified format.

#### 30051 **ENVIRONMENT VARIABLES**

30052 The following environment variables shall affect the execution of *prs*:

30053 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 30054 If *LANG* is unset or null, the corresponding value from the implementation-  
 30055 dependent default locale shall be used. If any of the internationalization variables  
 30056 contains an invalid setting, the utility shall behave as if none of the variables had  
 30057 been defined.

30058 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 30059 internationalization variables.

30060 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 30061 characters (for example, single-byte as opposed to multi-byte characters in  
 30062 arguments and input files).

#### 30063 *LC\_MESSAGES*

30064 Determine the locale that should be used to affect the format and contents of  
 30065 diagnostic messages written to standard error.

30066 *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

#### 30067 **ASYNCHRONOUS EVENTS**

30068 Default.

#### 30069 **STDOUT**

30070 The standard output shall be a text file whose format is dependent on the data keywords  
 30071 specified with the *-d* option.

#### 30072 **Data Keywords**

30073 Data keywords specify which parts of an SCCS file shall be retrieved and output. All parts of an  
 30074 SCCS file have an associated data keyword. A data keyword may appear in a *dataspec* multiple  
 30075 times.

30076 The information written by *prs* consists of:

- 30077 1. The user-supplied text
- 30078 2. Appropriate values (extracted from the SCCS file) substituted for the recognized data  
 30079 keywords in the order of appearance in the *dataspec*

30080 The format of a data keyword value is either simple (*'S'*), in which keyword substitution is  
 30081 direct, or multi-line (*'M'*).

30082 User-supplied text is any text other than recognized data keywords. A <tab> character is  
 30083 specified by *'\t'* and <newline> by *'\n'*. When the *-r* option is not specified, the default  
 30084 *dataspec* is:

30085 :PN::\n\n

30086 and the following *dataspec* is used for each selected delta:

30087 :Dt:\t:DL:\nMRs:\n:MR:COMMENTS:\n:C:\n

30088

30089

30090

30091

30092

30093

30094

30095

30096

30097

30098

30099

30100

30101

30102

30103

30104

30105

30106

30107

30108

30109

30110

30111

30112

30113

30114

30115

30116

30117

30118

30119

30120

30121

30122

30123

30124

30125

30126

30127

30128

30129

30130

30131

| SCCS File Data Keywords |                                                         |              |               |        |  |
|-------------------------|---------------------------------------------------------|--------------|---------------|--------|--|
| Keyword                 | Data Item                                               | File Section | Value         | Format |  |
| :Dt:                    | Delta information                                       | Delta Table  | See below*    | S      |  |
| :DL:                    | Delta line statistics                                   | "            | :Li:/Ld:/Lu:  | S      |  |
| :Li:                    | Lines inserted by Delta                                 | "            | nnnnn         | S      |  |
| :Ld:                    | Lines deleted by Delta                                  | "            | nnnnn         | S      |  |
| :Lu:                    | Lines unchanged by Delta                                | "            | nnnnn         | S      |  |
| :DT:                    | Delta type                                              | "            | D or R        | S      |  |
| :I:                     | SCCS ID string (SID)                                    | "            | See below**   | S      |  |
| :R:                     | Release number                                          | "            | nnnn          | S      |  |
| :L:                     | Level number                                            | "            | nnnn          | S      |  |
| :B:                     | Branch number                                           | "            | nnnn          | S      |  |
| :S:                     | Sequence number                                         | "            | nnnn          | S      |  |
| :D:                     | Date delta created                                      | "            | :Dy:/Dm:/Dd:  | S      |  |
| :Dy:                    | Year delta created                                      | "            | nn            | S      |  |
| :Dm:                    | Month delta created                                     | "            | nn            | S      |  |
| :Dd:                    | Day delta created                                       | "            | nn            | S      |  |
| :T:                     | Time delta created                                      | "            | :Th::Tm::Ts:  | S      |  |
| :Th:                    | Hour delta created                                      | "            | nn            | S      |  |
| :Tm:                    | Minutes delta created                                   | "            | nn            | S      |  |
| :Ts:                    | Seconds delta created                                   | "            | nn            | S      |  |
| :P:                     | Programmer who created Delta                            | "            | logname       | S      |  |
| :DS:                    | Delta sequence number                                   | "            | nnnn          | S      |  |
| :DP:                    | Predecessor Delta sequence number                       | "            | nnnn          | S      |  |
| :DI:                    | Sequence number of deltas included, excluded or ignored | "            | :Dn:/Dx:/Dg:  | S      |  |
| :Dn:                    | Deltas included (sequence #)                            | "            | :DS: :DS: ... | S      |  |
| :Dx:                    | Deltas excluded (sequence #)                            | "            | :DS: :DS: ... | S      |  |
| :Dg:                    | Deltas ignored (sequence #)                             | "            | :DS: :DS: ... | S      |  |
| :MR:                    | MR numbers for delta                                    | "            | text          | M      |  |
| :C:                     | Comments for delta                                      | "            | text          | M      |  |
| :UN:                    | User names                                              | User Names   | text          | M      |  |
| :FL:                    | Flag list                                               | Flags        | text          | M      |  |
| :Y:                     | Module type flag                                        | "            | text          | S      |  |
| :MF:                    | MR validation flag                                      | "            | yes or no     | S      |  |
| :MP:                    | MR validation program name                              | "            | text          | S      |  |
| :KF:                    | Keyword error, warning flag                             | "            | yes or no     | S      |  |
| :KV:                    | Keyword validation string                               | "            | text          | S      |  |
| :BF:                    | Branch flag                                             | "            | yes or no     | S      |  |
| :J:                     | Joint edit flag                                         | "            | yes or no     | S      |  |
| :LK:                    | Locked releases                                         | "            | :R: ...       | S      |  |
| :Q:                     | User-defined keyword                                    | "            | text          | S      |  |

30132

30133

30134

30135

30136

30137

30138

30139

30140

30141

30142

30143

30144

30145

30146

30147

| SCCS File Data Keywords |                              |              |                        |        |  |
|-------------------------|------------------------------|--------------|------------------------|--------|--|
| Keyword                 | Data Item                    | File Section | Value                  | Format |  |
| <b>:M:</b>              | Module name                  | "            | <i>text</i>            | S      |  |
| <b>:FB:</b>             | Floor boundary               | "            | <b>:R:</b>             | S      |  |
| <b>:CB:</b>             | Ceiling boundary             | "            | <b>:R:</b>             | S      |  |
| <b>:Ds:</b>             | Default SID                  | "            | <b>:I:</b>             | S      |  |
| <b>:ND:</b>             | Null delta flag              | "            | <b>yes or no</b>       | S      |  |
| <b>:FD:</b>             | File descriptive text        | Comments     | <i>text</i>            | M      |  |
| <b>:BD:</b>             | Body                         | Body         | <i>text</i>            | M      |  |
| <b>:GB:</b>             | Gotten body                  | "            | <i>text</i>            | M      |  |
| <b>:W:</b>              | A form of <i>what</i> string | N/A          | <b>:Z::M:\t:I:</b>     | S      |  |
| <b>:A:</b>              | A form of <i>what</i> string | N/A          | <b>:Z::Y::M::I::Z:</b> | S      |  |
| <b>:Z:</b>              | <i>what</i> string delimiter | N/A          | @( # )                 | S      |  |
| <b>:F:</b>              | SCCS file name               | N/A          | <i>text</i>            | S      |  |
| <b>:PN:</b>             | SCCS file path name          | N/A          | <i>text</i>            | S      |  |

30148

\* **:Dt::DT: :I: :D: :T: :P: :DS: :DP:**

30149

\*\* **:R::L::B::S:** if the delta is a branch delta (**:BF:= =yes**)

30150

**:R::L:** if the delta is not a branch delta (**:BF:= =no**)**30151 STDERR**

30152 Used only for diagnostic messages.

**30153 OUTPUT FILES**

30154 None.

**30155 EXTENDED DESCRIPTION**

30156 None.

**30157 EXIT STATUS**

30158 The following exit values shall be returned:

30159 0 Successful completion.

30160 &gt;0 An error occurred.

**30161 CONSEQUENCES OF ERRORS**

30162 Default.

**30163 APPLICATION USAGE**

30164 None.

**30165 EXAMPLES**

30166 1. The following example:

30167 prs -d "User Names for :F: are:\n:UN:" s.file

30168 may write to standard output:

30169 User Names for s.file are:

30170 xyz

30171 131

30172 abc

30173 2. The following example:

30174 prs -d "Delta for pgm :M:: :I: - :D: By :P:" -r s.file

30175 may write to standard output:  
 30176 Delta for pgm main.c: 3.7 – 77/12/01 By cas  
 30177 3. As a special case:  
 30178 prs s.file  
 30179 may write to standard output:  
 30180 s.file:  
 30181 <blank line>  
 30182 D 1.1 77/12/01 00:00:00 cas 1 000000/00000/00000  
 30183 MRs:  
 30184 b178–12345  
 30185 b179–54321  
 30186 COMMENTS:  
 30187 this is the comment line for s.file initial delta  
 30188 <blank line>  
 30189 for each delta table entry of the **D** type. The only option allowed to be used with this  
 30190 special case is the **–a** option.  
 30191 **RATIONALE**  
 30192 None.  
 30193 **FUTURE DIRECTIONS**  
 30194 A version of *prs* that fully supports the System Interface Definitions volume of  
 30195 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines may be introduced in a future  
 30196 issue.  
 30197 **SEE ALSO**  
 30198 *admin, delta, get, what*  
 30199 **CHANGE HISTORY**  
 30200 First released in Issue 2.  
 30201 **Issue 4**  
 30202 Format reorganized.  
 30203 Exceptions to Utility Syntax Guidelines conformance noted.  
 30204 Internationalized environment variable support mandated.  
 30205 **Issue 5**  
 30206 The phrase “in which keyword substitution is followed by a <newline>” is deleted from the end  
 30207 of the second paragraph of **Data Keywords** on page 783.  
 30208 The interpretation of the YY component of the **–c cutoff** argument is noted.

30209 **NAME**30210 `ps` — report process status30211 **SYNOPSIS**30212 UP XSI `ps [-aA][--defl][--G grouplist][--o format]...[-p proclist][-t termlist]`30213 `[-U userlist][--g grouplist][--n namelist][--u userlist]`

30214

30215 **DESCRIPTION**30216 The `ps` utility shall write information about processes, subject to having the appropriate  
30217 privileges to obtain information about those processes.30218 By default, `ps` selects all processes with the same effective user ID as the current user and the  
30219 same controlling terminal as the invoker.30220 **OPTIONS**30221 The `ps` utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
30222 Section 12.2, Utility Syntax Guidelines.

30223 The following options shall be supported:

30224 **-a** Write information for all processes associated with terminals. Implementations  
30225 may omit session leaders from this list.30226 **-A** Write information for all processes.30227 XSI **-d** Write information for all processes, except session leaders.30228 XSI **-e** Write information for all processes. (Equivalent to **-A**.)30229 XSI **-f** Generate a **full** listing. (See the STDOUT section for the contents of a **full** listing.)30230 XSI **-g grouplist** Write information for processes whose session leaders are given in *grouplist*. The  
30231 application shall ensure that the *grouplist* is a single argument in the form of a  
30232 <blank> or comma-separated list.30233 **-G grouplist** Write information for processes whose real group ID numbers are given in  
30234 *grouplist*. The application shall ensure that the *grouplist* is a single argument in the  
30235 form of a <blank> or comma-separated list.30236 XSI **-l** Generate a **long** listing. (See the STDOUT section for the contents of a **long** listing.)

30237

30238 XSI **-n namelist** Specify the name of an alternative system *namelist* file in place of the default. The  
30239 name of the default file and the format of a *namelist* file are unspecified.30240 **-o format** Write information according to the format specification given in *format*. This is  
30241 fully described in the STDOUT section. Multiple **-o** options can be specified; the  
30242 format specification shall be interpreted as the <space> character-separated  
30243 concatenation of all the *format* option-arguments.30244 **-p proclist** Write information for processes whose process ID numbers are given in *proclist*.  
30245 The application shall ensure that the *proclist* is a single argument in the form of a  
30246 <blank> or comma-separated list.30247 **-t termlist** Write information for processes associated with terminals given in *termlist*. The  
30248 application shall ensure that the *termlist* is a single argument in the form of a  
30249 XSI <blank> or comma-separated list. Terminal identifiers shall be given in one of two  
30250 forms: the device's file name (for example, **tty04**) or, if the device's file name starts  
30251 with **tty**, just the identifier following the characters **tty** (for example, "04").

30252 XSI     **-u *userlist***     Write information for processes whose user ID numbers or login names are given  
30253                                 in *userlist*. The application shall ensure that the *userlist* is a single argument in the  
30254                                 form of a <blank> or comma-separated list. In the listing, the numerical user ID is  
30255                                 written unless the **-f** option is used, in which case the login name is written.

30256             **-U *userlist***     Write information for processes whose real user ID numbers or login names are  
30257                                 given in *userlist*. The application shall ensure that the *userlist* is a single argument  
30258                                 in the form of a <blank> or comma-separated list.

30259             With the exception of **-o *format***, all of the options shown are used to select processes. If any are  
30260             specified, the default list shall be ignored and *ps* shall select the processes represented by the  
30261             bitwise-inclusive OR of all the selection-criteria options.

30262 **OPERANDS**

30263             None.

30264 **STDIN**

30265             Not used.

30266 **INPUT FILES**

30267             None.

30268 **ENVIRONMENT VARIABLES**

30269             The following environment variables shall affect the execution of *ps*:

30270             **COLUMNS**     Override the system-selected horizontal screen size, used to determine the number  
30271                                 of text columns to display. See the System Interface Definitions volume of  
30272                                 IEEE Std. 1003.1-200x, Chapter 8, Environment Variables for valid values and  
30273                                 results when it is unset or null.

30274             **LANG**             Provide a default value for the internationalization variables that are unset or null.  
30275                                 If *LANG* is unset or null, the corresponding value from the implementation-  
30276                                 dependent default locale shall be used. If any of the internationalization variables  
30277                                 contains an invalid setting, the utility shall behave as if none of the variables had  
30278                                 been defined.

30279             **LC\_ALL**            If set to a non-empty string value, override the values of all the other  
30280                                 internationalization variables.

30281             **LC\_CTYPE**        Determine the locale for the interpretation of sequences of bytes of text data as  
30282                                 characters (for example, single-byte as opposed to multi-byte characters in  
30283                                 arguments).

30284             **LC\_MESSAGES**     Determine the locale that should be used to affect the format and contents of  
30285                                 diagnostic messages written to standard error and informative messages written to  
30286                                 standard output.  
30287

30288 XSI     **NLSPATH**     Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

30289             **LC\_TIME**        Determine the format and contents of the date and time strings displayed.

30290 **ASYNCHRONOUS EVENTS**

30291             Default.

30292 **STDOUT**

30293             When the **-o** option is not specified, the standard output format is unspecified.

30294 XSI     On XSI-conformant systems, the output format is as follows. The column headings and  
30295             descriptions of the columns in a *ps* listing are given below. The precise meanings of these fields



are implementation-dependent. The letters 'f' and 'l' (below) indicate the option (**full** or **long**) that shall cause the corresponding heading to appear; **all** means that the heading always appears. Note that these two options determine only what information is provided for a process; they do not determine which processes are listed.

|       |              |       |                                                                                                   |
|-------|--------------|-------|---------------------------------------------------------------------------------------------------|
| 30300 | <b>F</b>     | (l)   | Flags (octal and additive) associated with the process.                                           |
| 30301 | <b>S</b>     | (l)   | The state of the process.                                                                         |
| 30302 | <b>UID</b>   | (f,l) | The user ID number of the process owner; the login name is printed under the <b>-f</b> option.    |
| 30303 |              |       |                                                                                                   |
| 30304 | <b>PID</b>   | (all) | The process ID of the process; it is possible to kill a process if this datum is known.           |
| 30305 |              |       |                                                                                                   |
| 30306 | <b>PPID</b>  | (f,l) | The process ID of the parent process.                                                             |
| 30307 | <b>C</b>     | (f,l) | Processor utilization for scheduling.                                                             |
| 30308 | <b>PRI</b>   | (l)   | The priority of the process; higher numbers mean lower priority.                                  |
| 30309 | <b>NI</b>    | (l)   | Nice value; used in priority computation.                                                         |
| 30310 | <b>ADDR</b>  | (l)   | The address of the process.                                                                       |
| 30311 | <b>SZ</b>    | (l)   | The size in blocks of the core image of the process.                                              |
| 30312 | <b>WCHAN</b> | (l)   | The event for which the process is waiting or sleeping; if blank, the process is running.         |
| 30313 |              |       |                                                                                                   |
| 30314 | <b>STIME</b> | (f)   | Starting time of the process.                                                                     |
| 30315 | <b>TTY</b>   | (all) | The controlling terminal for the process.                                                         |
| 30316 | <b>TIME</b>  | (all) | The cumulative execution time for the process.                                                    |
| 30317 | <b>CMD</b>   | (all) | The command name; the full command name and its arguments are written under the <b>-f</b> option. |
| 30318 |              |       |                                                                                                   |

A process that has exited and has a parent, but has not yet been waited for by the parent, is marked **defunct**.

Under the option **-f**, *ps* tries to determine the command name and arguments given when the process was created by examining memory or the swap area. Failing this, the command name, as it would appear without the option **-f**, is written in square brackets.

The **-o** option allows the output format to be specified under user control.

The application shall ensure that the format specification is a list of names presented as a single argument, <blank> or comma-separated. Each variable has a default header. The default header can be overridden by appending an equals sign and the new text of the header. The rest of the characters in the argument shall be used as the header text. The fields specified shall be written in the order specified on the command line, and should be arranged in columns in the output. The field widths shall be selected by the system to be at least as wide as the header text (default or overridden value). If the header text is null, such as **-o user=**, the field width shall be at least as wide as the default header text. If all header text fields are null, no header line shall be written.

The following names are recognized in the POSIX locale:

|       |               |                                                                                                                                                                    |
|-------|---------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 30335 | <b>ruser</b>  | The real user ID of the process. This shall be the textual user ID, if it can be obtained and the field width permits, or a decimal representation otherwise.      |
| 30336 |               |                                                                                                                                                                    |
| 30337 | <b>user</b>   | The effective user ID of the process. This shall be the textual user ID, if it can be obtained and the field width permits, or a decimal representation otherwise. |
| 30338 |               |                                                                                                                                                                    |
| 30339 | <b>rgroup</b> | The real group ID of the process. This shall be the textual group ID, if it can be obtained and the field width permits, or a decimal representation otherwise.    |
| 30340 |               |                                                                                                                                                                    |

|       |                                                                                                            |                                                                                                                   |
|-------|------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| 30341 | <b>group</b>                                                                                               | The effective group ID of the process. This shall be the textual group ID, if it can be                           |
| 30342 |                                                                                                            | obtained and the field width permits, or a decimal representation otherwise.                                      |
| 30343 | <b>pid</b>                                                                                                 | The decimal value of the process ID.                                                                              |
| 30344 | <b>ppid</b>                                                                                                | The decimal value of the parent process ID.                                                                       |
| 30345 | <b>pgid</b>                                                                                                | The decimal value of the process group ID.                                                                        |
| 30346 | <b>pcpu</b>                                                                                                | The ratio of CPU time used recently to CPU time available in the same period,                                     |
| 30347 |                                                                                                            | expressed as a percentage. The meaning of “recently” in this context is unspecified. The                          |
| 30348 |                                                                                                            | CPU time available is determined in an unspecified manner.                                                        |
| 30349 | <b>vsz</b>                                                                                                 | The size of the process in (virtual) memory in kilobytes as a decimal integer.                                    |
| 30350 | <b>nice</b>                                                                                                | The decimal value of the nice value of the process; see <i>nice</i> on page 698.                                  |
| 30351 | <b>etime</b>                                                                                               | In the POSIX locale, the elapsed time since the process was started, in the form:                                 |
| 30352 |                                                                                                            | [ [ <i>dd</i> -] <i>hh</i> : ] <i>mm</i> : <i>ss</i>                                                              |
| 30353 |                                                                                                            | where <i>dd</i> shall represent the number of days, <i>hh</i> the number of hours, <i>mm</i> the number           |
| 30354 |                                                                                                            | of minutes, and <i>ss</i> the number of seconds. The <i>dd</i> field shall be a decimal integer. The              |
| 30355 |                                                                                                            | <i>hh</i> , <i>mm</i> , and <i>ss</i> fields shall be two-digit decimal integers padded on the left with zeros.   |
| 30356 | <b>time</b>                                                                                                | In the POSIX locale, the cumulative CPU time of the process in the form:                                          |
| 30357 |                                                                                                            | [ [ <i>dd</i> -] <i>hh</i> : <i>mm</i> : <i>ss</i>                                                                |
| 30358 |                                                                                                            | The <i>dd</i> , <i>hh</i> , <i>mm</i> , and <i>ss</i> fields shall be as described in the <b>etime</b> specifier. |
| 30359 | <b>tty</b>                                                                                                 | The name of the controlling terminal of the process (if any) in the same format used by                           |
| 30360 |                                                                                                            | the <i>who</i> utility.                                                                                           |
| 30361 | <b>comm</b>                                                                                                | The name of the command being executed ( <i>argv</i> [0] value) as a string.                                      |
| 30362 | <b>args</b>                                                                                                | The command with all its arguments as a string. The implementation may truncate this                              |
| 30363 |                                                                                                            | value to the field width; it is implementation-dependent whether any further                                      |
| 30364 |                                                                                                            | truncation occurs. It is unspecified whether the string represented is a version of the                           |
| 30365 |                                                                                                            | argument list as it was passed to the command when it started, or is a version of the                             |
| 30366 |                                                                                                            | arguments as they may have been modified by the application. Applications cannot                                  |
| 30367 |                                                                                                            | depend on being able to modify their argument list and having that modification be                                |
| 30368 |                                                                                                            | reflected in the output of <i>ps</i> .                                                                            |
| 30369 | Any field need not be meaningful in all implementations. In such a case a hyphen (‘-’) should              |                                                                                                                   |
| 30370 | be output in place of the field value.                                                                     |                                                                                                                   |
| 30371 | Only <b>comm</b> and <b>args</b> shall be allowed to contain <blank> characters; all others shall not. Any |                                                                                                                   |
| 30372 | implementation-dependent variables shall be specified in the system documentation along with               |                                                                                                                   |
| 30373 | the default header and indicating if the field may contain <blank> characters.                             |                                                                                                                   |
| 30374 | The following table specifies the default header to be used in the POSIX locale corresponding to           |                                                                                                                   |
| 30375 | each format specifier.                                                                                     |                                                                                                                   |

Table 4-17 Variable Names and Default Headers in *ps*

| Format Specifier | Default Header | Format Specifier | Default Header |
|------------------|----------------|------------------|----------------|
| <b>args</b>      | <b>COMMAND</b> | <b>ppid</b>      | <b>PPID</b>    |
| <b>comm</b>      | <b>COMMAND</b> | <b>rgroup</b>    | <b>RGROUP</b>  |
| <b>etime</b>     | <b>ELAPSED</b> | <b>ruser</b>     | <b>RUSER</b>   |
| <b>group</b>     | <b>GROUP</b>   | <b>time</b>      | <b>TIME</b>    |
| <b>nice</b>      | <b>NI</b>      | <b>tty</b>       | <b>TT</b>      |
| <b>pcpu</b>      | <b>%CPU</b>    | <b>user</b>      | <b>USER</b>    |
| <b>pgid</b>      | <b>PGID</b>    | <b>vsz</b>       | <b>VSZ</b>     |
| <b>pid</b>       | <b>PID</b>     |                  |                |

**STDERR**

Used only for diagnostic messages.

**OUTPUT FILES**

None.

**EXTENDED DESCRIPTION**

None.

**EXIT STATUS**

The following exit values shall be returned:

0 Successful completion.

>0 An error occurred.

**CONSEQUENCES OF ERRORS**

Default.

**APPLICATION USAGE**

Things can change while *ps* is running; the snapshot it gives is only true for an instant, and might not be accurate by the time it is displayed.

The **args** format specifier is allowed to produce a truncated version of the command arguments. In some implementations, this information is no longer available when the *ps* utility is executed.

If the field width is too narrow to display a textual ID, the system may use a numeric version. Normally, the system would be expected to choose large enough field widths, but if a large number of fields were selected to write, it might squeeze fields to their minimum sizes to fit on one line. One way to ensure adequate width for the textual IDs is to override the default header for a field to make it larger than most or all user or group names.

There is no special quoting mechanism for header text. The header text is the rest of the argument. If multiple header changes are needed, multiple **-o** options can be used, such as:

```
ps -o "user=User Name" -o pid=Process\ ID
```

On some systems, especially multi-level secure systems, *ps* may be severely restricted and produce information only about child processes owned by the user.

Application writers should note that this utility need not be provided on systems that do not support the User Portability Utilities option.

**EXAMPLES**

The command:

```
ps -o user,pid,ppid=MOM -o args
```

30418 writes at least the following in the POSIX locale:

```
30419 USER PID MOM COMMAND
30420 helene 34 12 ps -o uid,pid,ppid=MOM -o args
```

30421 The contents of the **COMMAND** field need not be the same in all implementations, due to  
30422 possible truncation.

#### 30423 RATIONALE

30424 There is very little commonality between BSD and System V implementations of *ps*. Many  
30425 options conflict or have subtly different usages. The standard developers attempted to select a  
30426 set of options that were useful on a wide range of systems and selected options that either can be  
30427 implemented on both BSD and System V-based systems without breaking the current  
30428 implementations or where the options are sufficiently similar that any changes would not be  
30429 unduly problematic for users or implementors.

30430 It is recognized that on some systems, especially multi-level secure systems, *ps* may be nearly  
30431 useless. The default output has therefore been chosen such that it does not break historical  
30432 implementations and also is likely to provide at least some useful information on most systems.

30433 The major change is the addition of the format specification capability. The motivation for this  
30434 invention is to provide a mechanism for users to access a wider range of system information, if  
30435 the system permits it, in a portable manner. The fields chosen to appear in this volume of  
30436 IEEE Std. 1003.1-200x were arrived at after considering what concepts were likely to be both  
30437 reasonably useful to the “average” user and had a reasonable chance of being implemented on a  
30438 wide range of systems. Again it is recognized that not all systems are able to provide all the  
30439 information and, conversely, some may wish to provide more. It is hoped that the approach  
30440 adopted will be sufficiently flexible and extensible to accommodate most systems.  
30441 Implementations may be expected to introduce new format specifiers.

30442 The default output should consist of a short listing containing the process ID, terminal name,  
30443 cumulative execution time, and command name of each process.

30444 The preference of the standard developers would have been to make the format specification an  
30445 operand of the *ps* command. Unfortunately, BSD usage precluded this.

30446 At one time a format was included to display the environment array of the process. This was  
30447 deleted because there is no portable way to display it.

30448 The **-A** option is equivalent to the BSD **-g** and the SVID **-e**. Because the two systems differed, a  
30449 mnemonic compromise was selected.

30450 The **-a** option is described with some optional behavior because the SVID omits session leaders,  
30451 but BSD does not.

30452 In an early proposal, format specifiers appeared for priority and start time. The former was not  
30453 defined adequately in this volume of IEEE Std. 1003.1-200x and was removed in deference to the  
30454 defined nice value; the latter because elapsed time was considered to be more useful.

30455 In a new BSD version of *ps*, a **-O** option can be used to write all of the default information,  
30456 followed by additional format specifiers. This was not adopted because the default output is  
30457 implementation-dependent. Nevertheless, this is a useful option that should be reserved for that  
30458 purpose. In the **-o** option for the POSIX Shell and Utilities *ps*, the format is the concatenation of  
30459 each **-o**. Therefore, the user can have an alias or function that defines the beginning of their  
30460 desired format and add more fields to the end of the output in certain cases where that would be  
30461 useful.

30462 The format of the terminal name is unspecified, but the descriptions of *ps*, *talk*, *who*, and *write*  
30463 require that they all use the same format.

30464 The **pcpu** field indicates that the CPU time available is determined in an unspecified manner.  
30465 This is because it is difficult to express an algorithm that is useful across all possible machine  
30466 architectures. Historical counterparts to this value have attempted to show percentage of use in  
30467 the recent past, such as the preceding minute. Frequently, these values for all processes did not  
30468 add up to 100%. Implementations are encouraged to provide data in this field to users that will  
30469 help them identify processes currently affecting the performance of the system.

30470 **FUTURE DIRECTIONS**

30471 None.

30472 **SEE ALSO**

30473 *kill, nice, renice*

30474 **CHANGE HISTORY**

30475 First released in Issue 2.

30476 **Issue 4**

30477 Aligned with the ISO/IEC 9945-2:1993 standard.

30478 **Issue 6**

30479 This utility is now marked as part of the User Portability Utilities option. |

30480 The normative text is reworded to avoid use of the term “must” for application requirements. |

## 30481 NAME

30482       pwd — return working directory name

## 30483 SYNOPSIS

30484       pwd [-L | -P ]

## 30485 DESCRIPTION

30486       The *pwd* utility shall write to standard output an absolute path name of the current working  
 30487       directory, which does not contain the file names dot or dot-dot.

## 30488 OPTIONS

30489       The *pwd* utility shall conform to the System Interface Definitions volume of  
 30490       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

30491       The following options shall be supported by the implementation:

30492       **-L**       If the *PWD* environment variable contains an absolute path name of the current  
 30493       directory that does not contain the file names dot or dot-dot, *pwd* shall write this  
 30494       path name to standard output. Otherwise, the **-L** option shall behave as the **-P**  
 30495       option.

30496       **-P**       The absolute path name written shall not contain file names that, in the context of  
 30497       the path name, refer to files of type symbolic link.

30498       If both **-L** and **-P** are specified, the last one shall apply. If neither **-L** nor **-P** is specified, the *pwd*  
 30499       utility shall behave as if **-L** had been specified.

## 30500 OPERANDS

30501       None.

## 30502 STDIN

30503       Not used.

## 30504 INPUT FILES

30505       None.

## 30506 ENVIRONMENT VARIABLES

30507       The following environment variables shall affect the execution of *pwd*:

30508       **LANG**       Provide a default value for the internationalization variables that are unset or null.  
 30509       If *LANG* is unset or null, the corresponding value from the implementation-  
 30510       dependent default locale shall be used. If any of the internationalization variables  
 30511       contains an invalid setting, the utility shall behave as if none of the variables had  
 30512       been defined.

30513       **LC\_ALL**      If set to a non-empty string value, override the values of all the other  
 30514       internationalization variables.

30515       **LC\_MESSAGES**

30516       Determine the locale that should be used to affect the format and contents of  
 30517       diagnostic messages written to standard error.

30518 XSI       **NLS\_PATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

30519       **PWD**       If the **-P** option is in effect, this variable shall be set to an absolute path name of  
 30520       the current working directory that does not contain any components that specify  
 30521       symbolic links, does not contain any components that are dot, and does not  
 30522       contain any components that are dot-dot. If an application sets or unsets the value  
 30523       of *PWD*, the behavior of *pwd* is unspecified.

**30524 ASYNCHRONOUS EVENTS**

30525 Default.

**30526 STDOUT**

30527 The *pwd* utility output is an absolute path name of the current working directory:

30528 "%s\n", <directory pathname>

**30529 STDERR**

30530 Used only for diagnostic messages.

**30531 OUTPUT FILES**

30532 None.

**30533 EXTENDED DESCRIPTION**

30534 None.

**30535 EXIT STATUS**

30536 The following exit values shall be returned:

30537 0 Successful completion.

30538 >0 An error occurred.

**30539 CONSEQUENCES OF ERRORS**

30540 If an error is detected, output shall not be written to standard output, a diagnostic message shall  
30541 be written to standard error, and the exit status is not zero.

**30542 APPLICATION USAGE**

30543 None.

**30544 EXAMPLES**

30545 None.

**30546 RATIONALE**

30547 Some implementations have historically provided *pwd* as a shell special built-in command.

30548 In most utilities, if an error occurs, partial output may be written to standard output. This does  
30549 not happen in historical implementations of *pwd*. Because *pwd* is frequently used in historical  
30550 shell scripts without checking the exit status, it is important that the historical behavior is  
30551 required here; therefore, the CONSEQUENCES OF ERRORS section specifically disallows any  
30552 partial output being written to standard output.

**30553 FUTURE DIRECTIONS**

30554 None.

**30555 SEE ALSO**

30556 *cd*, the System Interfaces volume of IEEE Std. 1003.1-200x, *getcwd()*

**30557 CHANGE HISTORY**

30558 First released in Issue 2.

**30559 Issue 4**

30560 Aligned with the ISO/IEC 9945-2: 1993 standard.

**30561 Issue 6**

30562 The *-P* and *-L* options are added to describe actions relating to symbolic links as specified in the  
30563 IEEE P1003.2b draft standard.

## 30564 NAME

30565 qalter — alter batch job

## 30566 SYNOPSIS

```
30567 BE qalter [-a date_time][-A account_string][-c interval][-e path_name]
30568 [-h hold_list][-j join_list][-k keep_list][-l resource_list]
30569 [-m mail_options][-M mail_list][-N name][-o path_name]
30570 [-p priority][-r y|n][-S path_name_list][-u user_list]
30571 job_identifier ...
30572
```

## 30573 DESCRIPTION

30574 The attributes of a batch job are altered by a request to the batch server that manages the batch  
 30575 job. The *qalter* utility is a user-accessible batch client that requests the alteration of the attributes  
 30576 of one or more batch jobs.

30577 The *qalter* utility shall alter the attributes of those batch jobs, and only those batch jobs, for which  
 30578 a batch *job\_identifier* is presented to the utility.

30579 The *qalter* utility shall alter the attributes of batch jobs in the order in which the batch  
 30580 *job\_identifiers* are presented to the utility.

30581 If the *qalter* utility fails to process a batch *job\_identifier* successfully, the utility shall proceed to  
 30582 process the remaining batch *job\_identifiers*, if any.

30583 For each batch *job\_identifier* for which the *qalter* utility succeeds, each attribute of the identified  
 30584 batch job shall be altered as indicated by all the options presented to the utility.

30585 For each identified batch job for which the *qalter* utility fails, the utility shall not alter any  
 30586 attribute of the batch job.

30587 For each batch job that the *qalter* utility processes, the utility shall not modify any attribute other  
 30588 than those required by the options and option-arguments presented to the utility.

30589 The *qalter* utility shall alter batch jobs by sending a *Modify Job Request* to the batch server that  
 30590 manages each batch job. At the time the *qalter* utility exits, it shall have modified the batch job  
 30591 corresponding to each successfully processed batch *job\_identifier*. An attempt to alter the  
 30592 attributes of a batch job in the RUNNING state is implementation-dependent.

## 30593 OPTIONS

30594 The *qalter* utility shall conform to the System Interface Definitions volume of  
 30595 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

30596 The following options shall be supported by the implementation:

30597 **-a date\_time** Redefine the time at which the batch job becomes eligible for execution.

30598 The *qalter* utility shall accept an option-argument that conforms to the syntax of  
 30599 the *date\_time* operand of the *touch* utility.

30600 The *qalter* utility shall set the *Execution\_Time* attribute of the batch job to the  
 30601 number of seconds since the Epoch that is equivalent to the local time expressed  
 30602 by the value of the *date\_time* option-argument. Specifying a *date\_time* option-  
 30603 argument that represents a time (number of seconds since the Epoch) earlier than  
 30604 the time at which the utility exits shall have the same effect on batch job execution  
 30605 as if the **-a** option had not been presented to the utility. The Epoch is defined in the  
 30606 System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 3.151, Epoch.

30607 **-A account\_string**

30608 Redefine the account to which the resource consumption of the batch job should be



|       |                     |                                                                                                          |  |
|-------|---------------------|----------------------------------------------------------------------------------------------------------|--|
| 30609 |                     | charged.                                                                                                 |  |
| 30610 |                     | The syntax of the <i>account_string</i> option-argument is unspecified.                                  |  |
| 30611 |                     | The <i>qalter</i> utility shall set the <i>Account_Name</i> attribute of the batch job to the value      |  |
| 30612 |                     | of the <i>account_string</i> option-argument.                                                            |  |
| 30613 | <b>-c interval</b>  | Redefine whether the batch job should be checkpointed, and if so, how often.                             |  |
| 30614 |                     | The <i>qalter</i> utility shall accept a value for the interval option-argument that is one of           |  |
| 30615 |                     | the following:                                                                                           |  |
| 30616 | <i>n</i>            | No checkpointing is to be performed on the batch batch job                                               |  |
| 30617 |                     | (NO_CHECKPOINT).                                                                                         |  |
| 30618 | <i>s</i>            | Checkpointing is to be performed only when the batch server is shut                                      |  |
| 30619 |                     | down (CHECKPOINT_AT_SHUTDOWN).                                                                           |  |
| 30620 | <i>c</i>            | Automatic periodic checkpointing is to be performed at the                                               |  |
| 30621 |                     | <i>Minimum_Cpu_Interval</i> attribute of the batch queue, in units of CPU                                |  |
| 30622 |                     | minutes (CHECKPOINT_AT_MIN_CPU_INTERVAL).                                                                |  |
| 30623 | <i>c=minutes</i>    | Automatic periodic checkpointing is to be performed every <i>minutes</i>                                 |  |
| 30624 |                     | of CPU time, or every <i>Minimum_Cpu_Interval</i> minutes, whichever is                                  |  |
| 30625 |                     | greater. The <i>minutes</i> argument shall conform to the syntax for                                     |  |
| 30626 |                     | unsigned integers and shall be greater than zero.                                                        |  |
| 30627 |                     | An implementation may define other checkpoint intervals. The conformance                                 |  |
| 30628 |                     | document for an implementation shall describe any alternative checkpoint                                 |  |
| 30629 |                     | intervals, how they are specified, their internal behavior, and how they affect the                      |  |
| 30630 |                     | behavior of the utility.                                                                                 |  |
| 30631 |                     | The <i>qalter</i> utility shall set the <i>Checkpoint</i> attribute of the batch job to the value of the |  |
| 30632 |                     | <i>interval</i> option-argument.                                                                         |  |
| 30633 | <b>-e path_name</b> | Redefine the path to be used for the standard error stream of the batch job.                             |  |
| 30634 |                     | The <i>qalter</i> utility shall accept a <i>path_name</i> option-argument that conforms to the           |  |
| 30635 |                     | syntax of the <i>path_name</i> element defined in the POSIX.1-1990 standard, which can                   |  |
| 30636 |                     | be preceded by a host name element of the form <i>hostname</i> :                                         |  |
| 30637 |                     | If the <i>path_name</i> option-argument constitutes an absolute path name, the <i>qalter</i>             |  |
| 30638 |                     | utility shall set the <i>Error_Path</i> attribute of the batch job to the value of the                   |  |
| 30639 |                     | <i>path_name</i> option-argument, including the host name element, if present.                           |  |
| 30640 |                     | If the <i>path_name</i> option-argument constitutes a relative path name and no host                     |  |
| 30641 |                     | name element is specified, the <i>qalter</i> utility shall set the <i>Error_Path</i> attribute of the    |  |
| 30642 |                     | batch job to the value of the absolute path name derived by expanding the                                |  |
| 30643 |                     | <i>path_name</i> option-argument relative to the current directory of the process that                   |  |
| 30644 |                     | executes the <i>qalter</i> utility.                                                                      |  |
| 30645 |                     | If the <i>path_name</i> option-argument constitutes a relative path name and a host name                 |  |
| 30646 |                     | element is specified, the <i>qalter</i> utility shall set the <i>Error_Path</i> attribute of the batch   |  |
| 30647 |                     | job to the value of the option-argument without expansion.                                               |  |
| 30648 |                     | If the <i>path_name</i> option-argument does not include a host name element, the <i>qalter</i>          |  |
| 30649 |                     | utility shall prefix the path name in the <i>Error_Path</i> attribute with <i>hostname</i> :, where      |  |
| 30650 |                     | <i>hostname</i> is the name of the host upon which the <i>qalter</i> utility is being executed.          |  |
| 30651 | <b>-h hold_list</b> | Redefine the types of holds, if any, on the batch job. The <i>qalter</i> <b>-h</b> option shall          |  |
| 30652 |                     | accept a value for the <i>hold_list</i> option-argument that is a string of alphanumeric                 |  |

|       |                     |                                                                                                             |
|-------|---------------------|-------------------------------------------------------------------------------------------------------------|
| 30653 |                     | characters in the portable character set (see the System Interface Definitions                              |
| 30654 |                     | volume of IEEE Std. 1003.1-200x, Section 6.1, Portable Character Set).                                      |
| 30655 |                     | The <i>qalter</i> utility shall accept a value for the <i>hold_list</i> option-argument that is a           |
| 30656 |                     | string of one or more of the characters 'u', 's', or 'o', or the single character                           |
| 30657 |                     | 'n'. For each unique character in the <i>hold_list</i> option-argument, the <i>qalter</i> utility           |
| 30658 |                     | shall add a value to the <i>Hold_Types</i> attribute of the batch job as follows, each                      |
| 30659 |                     | representing a different hold type:                                                                         |
| 30660 |                     | <b>u</b> USER                                                                                               |
| 30661 |                     | <b>s</b> SYSTEM                                                                                             |
| 30662 |                     | <b>o</b> OPERATOR                                                                                           |
| 30663 |                     | If any of these characters are duplicated in the <i>hold_list</i> option-argument, the                      |
| 30664 |                     | duplicates shall be ignored. An existing <i>Hold_Types</i> attribute can be cleared by the                  |
| 30665 |                     | hold type:                                                                                                  |
| 30666 |                     | <b>n</b> NO_HOLD                                                                                            |
| 30667 |                     | The <i>qalter</i> utility shall consider it an error if any hold type other than <b>n</b> is combined       |
| 30668 |                     | with hold type <b>n</b> . Strictly conforming applications shall not repeat any of the                      |
| 30669 |                     | characters 'u', 's', 'o', or 'n' within the <i>hold_list</i> option-argument. The <i>qalter</i>             |
| 30670 |                     | utility shall permit the repetition of characters, but shall not assign additional                          |
| 30671 |                     | meaning to the repeated characters. An implementation may define other hold                                 |
| 30672 |                     | types. The conformance document for an implementation shall describe any                                    |
| 30673 |                     | additional hold types, how they are specified, their internal behavior, and how                             |
| 30674 |                     | they affect the behavior of the utility.                                                                    |
| 30675 | <b>-j join_list</b> | Redefine which streams of the batch job are to be merged. The <i>qalter</i> <b>-j</b> option shall          |
| 30676 |                     | accept a value for the <i>join_list</i> option-argument that is a string of alphanumeric                    |
| 30677 |                     | characters in the portable character set (see the System Interface Definitions                              |
| 30678 |                     | volume of IEEE Std. 1003.1-200x, Section 6.1, Portable Character Set).                                      |
| 30679 |                     | The <i>qalter</i> utility shall accept a <i>join_list</i> option-argument that consists of one or           |
| 30680 |                     | more of the characters 'e' and 'o', or the single character 'n'.                                            |
| 30681 |                     | All of the other batch job output streams specified shall be merged into the output                         |
| 30682 |                     | stream represented by the character listed first in the <i>join_list</i> option-argument.                   |
| 30683 |                     | For each unique character in the <i>join_list</i> option-argument, the <i>qalter</i> utility shall          |
| 30684 |                     | add a value to the <i>Join_Path</i> attribute of the batch job as follows, each representing                |
| 30685 |                     | a different batch job stream to join:                                                                       |
| 30686 |                     | <b>e</b> The standard error of the batch batch job (JOIN_STD_ERROR).                                        |
| 30687 |                     | <b>o</b> The standard output of the batch batch job (JOIN_STD_OUTPUT).                                      |
| 30688 |                     | An existing <i>Join_Path</i> attribute can be cleared by the join type:                                     |
| 30689 |                     | <b>n</b> NO_JOIN                                                                                            |
| 30690 |                     | If <b>n</b> is specified, then no files are joined. The <i>qalter</i> utility shall consider it an error if |
| 30691 |                     | any join type other than <b>n</b> is combined with join type <b>n</b> .                                     |
| 30692 |                     | Strictly conforming applications shall not repeat any of the characters 'e', 'o', or                        |
| 30693 |                     | 'n' within the <i>join_list</i> option-argument. The <i>qalter</i> utility shall permit the                 |
| 30694 |                     | repetition of characters, but shall not assign additional meaning to the repeated                           |
| 30695 |                     | characters.                                                                                                 |

|       |                         |                                                                                                            |
|-------|-------------------------|------------------------------------------------------------------------------------------------------------|
| 30696 |                         | An implementation may define other join types. The conformance document for an                             |
| 30697 |                         | implementation shall describe any additional batch job streams, how they are                               |
| 30698 |                         | specified, their internal behavior, and how they affect the behavior of the utility.                       |
| 30699 | <b>-k keep_list</b>     | Redefine which output of the batch job to retain on the execution host.                                    |
| 30700 |                         | The <i>qalter</i> <b>-k</b> option shall accept a value for the <i>keep_list</i> option-argument that is a |
| 30701 |                         | string of alphanumeric characters in the portable character set (see the System                            |
| 30702 |                         | Interface Definitions volume of IEEE Std. 1003.1-200x, Section 6.1, Portable                               |
| 30703 |                         | Character Set).                                                                                            |
| 30704 |                         | The <i>qalter</i> utility shall accept a <i>keep_list</i> option-argument that consists of one or          |
| 30705 |                         | more of the characters 'e' and 'o' or the single character 'n'.                                            |
| 30706 |                         | For each unique character in the <i>keep_list</i> option-argument, the <i>qalter</i> utility shall         |
| 30707 |                         | add a value to the <i>Keep_Files</i> attribute of the batch job as follows, each representing              |
| 30708 |                         | a different batch job stream to keep:                                                                      |
| 30709 | <b>e</b>                | The standard error of the batch batch job (KEEP_STD_ERROR).                                                |
| 30710 | <b>o</b>                | The standard output of the batch batch job (KEEP_STD_OUTPUT).                                              |
| 30711 |                         | If both 'e' and 'o' are specified, then both files are retained. An existing                               |
| 30712 |                         | <i>Keep_Files</i> attribute can be cleared by the keep type:                                               |
| 30713 | <b>n NO_KEEP</b>        |                                                                                                            |
| 30714 |                         | If <b>n</b> is specified, then no files are retained. The <i>qalter</i> utility shall consider it an error |
| 30715 |                         | if any keep type other than <b>n</b> is combined with keep type <b>n</b> .                                 |
| 30716 |                         | Strictly conforming applications shall not repeat any of the characters 'e', 'o', or                       |
| 30717 |                         | 'n' within the <i>keep_list</i> option-argument. The <i>qalter</i> utility shall permit the                |
| 30718 |                         | repetition of characters, but shall not assign additional meaning to the repeated                          |
| 30719 |                         | characters. An implementation may define other keep types. The conformance                                 |
| 30720 |                         | document for an implementation shall describe any additional keep types, how                               |
| 30721 |                         | they are specified, their internal behavior, and how they affect the behavior of the                       |
| 30722 |                         | utility.                                                                                                   |
| 30723 | <b>-l resource_list</b> |                                                                                                            |
| 30724 |                         | Redefine the resources that are allowed or required by the batch job.                                      |
| 30725 |                         | The <i>qalter</i> utility shall accept a <i>resource_list</i> option-argument that conforms to the         |
| 30726 |                         | following syntax:                                                                                          |
| 30727 |                         | resource=value[, ,resource=value, ,...]                                                                    |
| 30728 |                         | The <i>qalter</i> utility shall set one entry in the value of the <i>Resource_List</i> attribute of the    |
| 30729 |                         | batch job for each resource listed in the <i>resource_list</i> option-argument.                            |
| 30730 |                         | Because the list of supported resource names might vary by batch server, the <i>qalter</i>                 |
| 30731 |                         | utility shall rely on the batch server to validate the resource names and associated                       |
| 30732 |                         | values. See Section 3.3.3 on page 157 for a means of removing <i>keyword=value</i> (and                    |
| 30733 |                         | <i>value@keyword</i> ) pairs and other general rules for list-oriented batch job attributes.               |
| 30734 | <b>-m mail_options</b>  |                                                                                                            |
| 30735 |                         | Redefine the points in the execution of the batch job at which the batch server is to                      |
| 30736 |                         | send mail about a change in the state of the batch job.                                                    |
| 30737 |                         | The <i>qalter</i> <b>-m</b> option shall accept a value for the <i>mail_options</i> option-argument that   |
| 30738 |                         | is a string of alphanumeric characters in the portable character set (see the System                       |
| 30739 |                         | Interface Definitions volume of IEEE Std. 1003.1-200x, Section 6.1, Portable                               |

|       |                            |                                                                                                          |
|-------|----------------------------|----------------------------------------------------------------------------------------------------------|
| 30740 |                            | Character Set).                                                                                          |
| 30741 |                            | The <i>qalter</i> utility shall accept a value for the <i>mail_options</i> option-argument that is a     |
| 30742 |                            | string of one or more of the characters 'e', 'b', and 'a', or the single character                       |
| 30743 |                            | 'n'. For each unique character in the <i>mail_options</i> option-argument, the <i>qalter</i>             |
| 30744 |                            | utility shall add a value to the <i>Mail_Users</i> attribute of the batch job as follows, each           |
| 30745 |                            | representing a different time during the life of a batch job at which to send mail:                      |
| 30746 | <i>e</i>                   | MAIL_AT_EXIT                                                                                             |
| 30747 | <i>b</i>                   | MAIL_AT_BEGINNING                                                                                        |
| 30748 | <i>a</i>                   | MAIL_AT_ABORT                                                                                            |
| 30749 |                            | If any of these characters are duplicated in the <i>mail_options</i> option-argument, the                |
| 30750 |                            | duplicates shall be ignored.                                                                             |
| 30751 |                            | An existing <i>Mail_Points</i> attribute can be cleared by the mail type:                                |
| 30752 | <i>n</i>                   | NO_MAIL                                                                                                  |
| 30753 |                            | If <i>n</i> is specified, then mail is not sent. The <i>qalter</i> utility shall consider it an error if |
| 30754 |                            | any mail type other than <i>n</i> is combined with mail type <i>n</i> . Strictly conforming              |
| 30755 |                            | applications shall not repeat any of the characters 'e', 'b', 'a', or 'n' within                         |
| 30756 |                            | the <i>mail_options</i> option-argument. The <i>qalter</i> utility shall permit the repetition of        |
| 30757 |                            | characters but shall not assign additional meaning to the repeated characters.                           |
| 30758 |                            | An implementation may define other mail types. The conformance document for                              |
| 30759 |                            | an implementation shall describe any additional mail types, how they are                                 |
| 30760 |                            | specified, their internal behavior, and how they affect the behavior of the utility.                     |
| 30761 | <b>-M</b> <i>mail_list</i> | Redefine the list of users to which the batch server that executes the batch job is to                   |
| 30762 |                            | send mail, if the batch server sends mail about the batch job.                                           |
| 30763 |                            | The syntax of the <i>mail_list</i> option-argument is unspecified. If the implementation                 |
| 30764 |                            | of the <i>qalter</i> utility uses a name service to locate users, the utility shall accept the           |
| 30765 |                            | syntax used by the name service.                                                                         |
| 30766 |                            | If the implementation of the <i>qalter</i> utility does not use a name service to locate                 |
| 30767 |                            | users, the implementation shall accept the following syntax for user names:                              |
| 30768 |                            | mail_address[, ,mail_address, , ...]                                                                     |
| 30769 |                            | The interpretation of <i>mail_address</i> is implementation-dependent.                                   |
| 30770 |                            | The <i>qalter</i> utility shall set the <i>Mail_Users</i> attribute of the batch job to the value of     |
| 30771 |                            | the <i>mail_list</i> option-argument.                                                                    |
| 30772 | <b>-N</b> <i>name</i>      | Redefine the name of the batch job.                                                                      |
| 30773 |                            | The <i>qalter</i> <b>-N</b> option shall accept a value for the <i>name</i> option argument that is a    |
| 30774 |                            | string of up to 15 alphanumeric characters in the portable character set (see the                        |
| 30775 |                            | System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 6.1, Portable                      |
| 30776 |                            | Character Set) where the first character is alphabetic.                                                  |
| 30777 |                            | The syntax of the <i>name</i> option-argument is unspecified.                                            |
| 30778 |                            | The <i>qalter</i> utility shall set the <i>Job_Name</i> attribute of the batch job to the value of the   |
| 30779 |                            | <i>name</i> option-argument.                                                                             |
| 30780 | <b>-o</b> <i>path_name</i> | Redefine the path for the standard output of the batch job.                                              |

|       |                          |                                                                                                            |
|-------|--------------------------|------------------------------------------------------------------------------------------------------------|
| 30781 |                          | The <i>qalter</i> utility shall accept a <i>path_name</i> option-argument that conforms to the             |
| 30782 |                          | syntax of the <i>path_name</i> element defined in the POSIX.1-1990 standard, which can                     |
| 30783 |                          | be preceded by a host name element of the form <i>hostname</i> ..                                          |
| 30784 |                          | If the <i>path_name</i> option-argument constitutes an absolute path name, the <i>qalter</i>               |
| 30785 |                          | utility shall set the <i>Output_Path</i> attribute of the batch job to the value of the                    |
| 30786 |                          | <i>path_name</i> option-argument.                                                                          |
| 30787 |                          | If the <i>path_name</i> option-argument constitutes a relative path name and no host                       |
| 30788 |                          | name element is specified, the <i>qalter</i> utility shall set the <i>Output_Path</i> attribute of the     |
| 30789 |                          | batch job to the absolute path name derived by expanding the <i>path_name</i> option-                      |
| 30790 |                          | argument relative to the current directory of the process that executes the <i>qalter</i>                  |
| 30791 |                          | utility.                                                                                                   |
| 30792 |                          | If the <i>path_name</i> option-argument constitutes a relative path name and a host name                   |
| 30793 |                          | element is specified, the <i>qalter</i> utility shall set the <i>Output_Path</i> attribute of the batch    |
| 30794 |                          | job to option-argument without any expansion of the path name.                                             |
| 30795 |                          | If the <i>path_name</i> option-argument does not include a host name element, the <i>qalter</i>            |
| 30796 |                          | utility shall prefix the path name in the <i>Output_Path</i> attribute with <i>hostname</i> .,             |
| 30797 |                          | where <i>hostname</i> is the name of the host upon which the <i>qalter</i> utility is being                |
| 30798 |                          | executed.                                                                                                  |
| 30799 | <b>-p priority</b>       | Redefine the priority of the batch job.                                                                    |
| 30800 |                          | The <i>qalter</i> utility shall accept a value for the priority option-argument that                       |
| 30801 |                          | conforms to the syntax for signed decimal integers, and which is not less than                             |
| 30802 |                          | -1 024 and not greater than 1 023.                                                                         |
| 30803 |                          | The <i>qalter</i> utility shall set the <i>Priority</i> attribute of the batch job to the value of the     |
| 30804 |                          | <i>priority</i> option-argument.                                                                           |
| 30805 | <b>-r y   n</b>          | Redefine whether the batch job is rerunnable.                                                              |
| 30806 |                          | If the value of the option-argument is <i>y</i> , the <i>qalter</i> utility shall set the <i>Rerunable</i> |
| 30807 |                          | attribute of the batch job to TRUE.                                                                        |
| 30808 |                          | If the value of the option-argument is <i>n</i> , the <i>qalter</i> utility shall set the <i>Rerunable</i> |
| 30809 |                          | attribute of the batch job to FALSE.                                                                       |
| 30810 |                          | The <i>qalter</i> utility shall consider it an error if any character other than 'y' or 'n' is             |
| 30811 |                          | specified in the option-argument.                                                                          |
| 30812 | <b>-S path_name_list</b> |                                                                                                            |
| 30813 |                          | Redefine the shell that interprets the script at the destination system.                                   |
| 30814 |                          | The <i>qalter</i> utility shall accept a <i>path_name_list</i> option-argument that conforms to            |
| 30815 |                          | the following syntax:                                                                                      |
| 30816 |                          | pathname[@host][, , pathname[@host] , , . . . ]                                                            |
| 30817 |                          | The <i>qalter</i> utility shall accept only one path name that is missing a corresponding                  |
| 30818 |                          | host name. The <i>qalter</i> utility shall allow only one path name per named host.                        |
| 30819 |                          | The <i>qalter</i> utility shall add a value to the <i>Shell_Path_List</i> attribute of the batch job       |
| 30820 |                          | for each entry in the <i>path_name_list</i> option-argument. See Section 3.3.3 on page 157                 |
| 30821 |                          | for a means of removing <i>keyword=value</i> (and <i>value@keyword</i> ) pairs and other                   |
| 30822 |                          | general rules for list-oriented batch job attributes.                                                      |
| 30823 | <b>-u user_list</b>      | Redefine the user name under which the batch job is to run at the destination                              |
| 30824 |                          | system.                                                                                                    |

|       |                              |                                                                                                         |
|-------|------------------------------|---------------------------------------------------------------------------------------------------------|
| 30825 |                              | The <i>qalter</i> utility shall accept a <i>user_list</i> option-argument that conforms to the          |
| 30826 |                              | following syntax:                                                                                       |
| 30827 |                              | <code>username[@host][, ,username[@host], , ...]</code>                                                 |
| 30828 |                              | The <i>qalter</i> utility shall accept only one user name that is missing a corresponding               |
| 30829 |                              | host name. The <i>qalter</i> utility shall accept only one user name per named host.                    |
| 30830 |                              | The <i>qalter</i> utility shall add a value to the <i>User_List</i> attribute of the batch job for each |
| 30831 |                              | entry in the <i>user_list</i> option-argument. See Section 3.3.3 on page 157 for a means of             |
| 30832 |                              | removing <i>keyword=value</i> (and <i>value@keyword</i> ) pairs and other general rules for             |
| 30833 |                              | list-oriented batch job attributes.                                                                     |
| 30834 | <b>OPERANDS</b>              |                                                                                                         |
| 30835 |                              | The <i>qalter</i> utility shall accept one or more operands that conform to the syntax for a batch      |
| 30836 |                              | <i>job_identifier</i> (see Section 3.3.1 on page 156).                                                  |
| 30837 | <b>STDIN</b>                 |                                                                                                         |
| 30838 |                              | Not used.                                                                                               |
| 30839 | <b>INPUT FILES</b>           |                                                                                                         |
| 30840 |                              | None.                                                                                                   |
| 30841 | <b>ENVIRONMENT VARIABLES</b> |                                                                                                         |
| 30842 |                              | The following environment variables shall affect the execution of <i>qalter</i> :                       |
| 30843 | <i>LANG</i>                  | Provide a default value for the internationalization variables that are unset or null.                  |
| 30844 |                              | If <i>LANG</i> is unset or null, the corresponding value from the implementation-                       |
| 30845 |                              | dependent default locale shall be used. If any of the internationalization variables                    |
| 30846 |                              | contains an invalid setting, the utility shall behave as if none of the variables had                   |
| 30847 |                              | been defined.                                                                                           |
| 30848 | <i>LC_ALL</i>                | If set to a non-empty string value, override the values of all the other                                |
| 30849 |                              | internationalization variables.                                                                         |
| 30850 | <i>LC_CTYPE</i>              | Determine the locale for the interpretation of sequences of bytes of text data as                       |
| 30851 |                              | characters (for example, single-byte as opposed to multi-byte characters in                             |
| 30852 |                              | arguments).                                                                                             |
| 30853 | <i>LC_MESSAGES</i>           |                                                                                                         |
| 30854 |                              | Determine the locale that should be used to affect the format and contents of                           |
| 30855 |                              | diagnostic messages written to standard error.                                                          |
| 30856 | <i>LC_TIME</i>               | Determine the format and contents of date and time strings written by <i>qalter</i> .                   |
| 30857 | <i>LOGNAME</i>               | Determine the login name of the user.                                                                   |
| 30858 | <i>TZ</i>                    | Determine the timezone in which the time and date are written. If the <i>TZ</i> variable                |
| 30859 |                              | is not set, an unspecified system default timezone is used.                                             |
| 30860 | <b>ASYNCHRONOUS EVENTS</b>   |                                                                                                         |
| 30861 |                              | Default.                                                                                                |
| 30862 | <b>STDOUT</b>                |                                                                                                         |
| 30863 |                              | None.                                                                                                   |
| 30864 | <b>STDERR</b>                |                                                                                                         |
| 30865 |                              | Used only for diagnostic messages.                                                                      |

30866 **OUTPUT FILES**

30867 None.

30868 **EXTENDED DESCRIPTION**

30869 None.

30870 **EXIT STATUS**

30871 The following exit values shall be returned:

30872 0 Successful completion.

30873 &gt;0 An error occurred.

30874 **CONSEQUENCES OF ERRORS**

30875 In addition to the default behavior, the *qalter* utility shall not be required to write a diagnostic  
 30876 message to standard error when the error reply received from a batch server indicates that the  
 30877 batch *job\_identifier* does not exist on the server. Whether or not the *qalter* utility attempts to  
 30878 locate the batch job on other batch servers is implementation-dependent.

30879 **APPLICATION USAGE**

30880 None.

30881 **EXAMPLES**

30882 None.

30883 **RATIONALE**30884 The *qalter* utility allows users to change the attributes of a batch job.

30885 As a means of altering a queued job, the *qalter* utility is superior to deleting and requeuing the  
 30886 batch job insofar as an altered job retains its place in the queue with some traditional selection  
 30887 algorithms. In addition, the *qalter* utility is both shorter and simpler than a sequence of *qdel* and  
 30888 *qsub* utilities.

30889 The result of an attempt on the part of a user to alter a batch job in a RUNNING state is  
 30890 implementation-dependent because a batch job in the RUNNING state will already have opened  
 30891 its output files and otherwise performed any actions indicated by the options in effect at the  
 30892 time the batch job began execution.

30893 The options processed by the *qalter* utility are identical to those of the *qsub* utility, with a few  
 30894 exceptions: *-V*, *-v*, and *-q*. The *-V* and *-v* are inappropriate for the *qalter* utility, since they  
 30895 capture potentially transient environment information from the submitting process. The *-q*  
 30896 option would specify a new queue, which would largely negate the previously stated advantage  
 30897 of using *qalter*; furthermore, the *qmove* utility provides a superior means of moving jobs.

30898 Each of the following paragraphs provides the rationale for a *qalter* option.30899 Additional rationale concerning these options can be found in the rationale for the *qsub* utility.

30900 The *-a* option allows users to alter the date and time at which a batch job becomes eligible to  
 30901 run.

30902 The *-A* option allows users to change the account that will be charged for the resources  
 30903 consumed by the batch job. Support for the *-A* option is mandatory for conforming  
 30904 implementations of *qalter*, even though support of accounting is optional for servers. Whether or  
 30905 not to support accounting is left to the implementor of the server, but mandatory support of the  
 30906 *-A* option assures users of a consistent interface and allows them to control accounting on  
 30907 servers that support accounting.

30908 The *-c* option allows users to alter the checkpointing interval of a batch job. A checkpointing  
 30909 system, which is not defined by IEEE Std. 1003.1-200x, allows recovery of a batch job at the most

30910 recent checkpoint in the event of a crash. Checkpointing is typically used for jobs that consume  
 30911 expensive computing time or must meet a critical schedule. Users should be allowed to make  
 30912 the tradeoff between the overhead of checkpointing and the risk to the timely completion of the  
 30913 batch job; therefore, this volume of IEEE Std. 1003.1-200x provides the checkpointing interval  
 30914 option. Support for checkpointing is optional for servers.

30915 The **-e** option allows users to alter the name and location of the standard error stream written by  
 30916 a batch job. However, the path of the standard error stream is meaningless if the value of the  
 30917 *Join\_Path* attribute of the batch job is TRUE.

30918 The **-h** option allows users to set the hold type in the *Hold\_Types* attribute of a batch job. The  
 30919 *qhold* and *qrls* utilities add or remove hold types to the *Hold\_Types* attribute, respectively. The **-h**  
 30920 option has been modified to allow for implementation-dependent hold types.

30921 The **-j** option allows users to alter the decision to join (merge) the standard error stream of the  
 30922 batch job with the standard output stream of the batch job.

30923 The **-l** option allows users to change the resource limits imposed on a batch job.

30924 The **-m** option allows users to modify the list of points in the life of a batch job at which the  
 30925 designated users will receive mail notification.

30926 The **-M** option allows users to alter the list of users who will receive notification about events in  
 30927 the life of a batch job.

30928 The **-N** option allows users to change the name of a batch job.

30929 The **-o** option allows users to alter the name and path to which the standard output stream of  
 30930 the batch job will be written.

30931 The **-P** option allows users to modify the priority of a batch job. Support for priority is optional  
 30932 for batch servers.

30933 The **-r** option allows users to alter the rerunability status of a batch job.

30934 The **-S** option allows users to change the name and location of the shell image that will be  
 30935 invoked to interpret the script of the batch job. This option has been modified to allow a list of  
 30936 shell name and locations associated with different host.

30937 The **-u** option allows users to change the user identifier under which the batch job will execute.

30938 As with other batch utilities, implementors can extend the *qalter* utility using the **-W** option.

30939 The *job\_identifier* operand syntax is provided so that the user can differentiate between the  
 30940 originating and destination (or executing) batch server. These may or may not be the same. The  
 30941 *.server\_name* portion identifies the originating batch server, while the *@server* portion identifies  
 30942 the destination batch server.

30943 Historically, the *qalter* utility has been a component of the Network Queuing System (NQS), the  
 30944 existing practice from which this utility has been derived.

#### 30945 FUTURE DIRECTIONS

30946 None.

#### 30947 SEE ALSO

30948 *qdel*, *qhold*, *qmove*, *qrls*, *qsub*, *touch*, Chapter 3 on page 133

#### 30949 CHANGE HISTORY

30950 Derived from IEEE Std. 1003.2d-1994.



30951 **NAME**

30952           qdel — delete batch jobs

30953 **SYNOPSIS**30954 BE       qdel *job\_identifier* ...

30955

30956 **DESCRIPTION**

30957       A batch job is deleted by sending a request to the batch server that manages the batch job. A  
 30958       batch job that has been deleted is no longer subject to management by batch services.

30959       The *qdel* utility is a user-accessible client of batch services that requests the deletion of one or  
 30960       more batch jobs.

30961       The *qdel* utility shall request a batch server to delete those batch jobs for which a batch  
 30962       *job\_identifier* is presented to the utility.

30963       The *qdel* utility shall delete batch jobs in the order in which their batch *job\_identifiers* are  
 30964       presented to the utility.

30965       If the *qdel* utility fails to process any batch *job\_identifier* successfully, the utility shall proceed to  
 30966       process the remaining batch *job\_identifiers*, if any.

30967       The *qdel* utility shall delete each batch job by sending a *Delete Job Request* to the batch server that  
 30968       manages the batch job.

30969       The *qdel* utility shall not exit until the batch job corresponding to each successfully processed  
 30970       batch *job\_identifier* has been deleted.

30971 **OPTIONS**

30972       None.

30973 **OPERANDS**

30974       The *qdel* utility shall accept one or more operands that conform to the syntax for a batch  
 30975       *job\_identifier* (see Section 3.3.1 on page 156).

30976 **STDIN**

30977       Not used.

30978 **INPUT FILES**

30979       None.

30980 **ENVIRONMENT VARIABLES**30981       The following environment variables shall affect the execution of *qdel*:

30982       *LANG*       Provide a default value for the internationalization variables that are unset or null.  
 30983       If *LANG* is unset or null, the corresponding value from the implementation-  
 30984       dependent default locale shall be used. If any of the internationalization variables  
 30985       contains an invalid setting, the utility shall behave as if none of the variables had  
 30986       been defined.

30987       *LC\_ALL*      If set to a non-empty string value, override the values of all the other  
 30988       internationalization variables.

30989       *LC\_CTYPE*   Determine the locale for the interpretation of sequences of bytes of text data as  
 30990       characters (for example, single-byte as opposed to multi-byte characters in  
 30991       arguments).

30992       *LC\_MESSAGES*

30993       Determine the locale that should be used to affect the format and contents of  
 30994       diagnostic messages written to standard error.

|       |                               |                                                                                                                      |
|-------|-------------------------------|----------------------------------------------------------------------------------------------------------------------|
| 30995 | <i>LC_TIME</i>                | Determine the format and contents of date and time strings written by <i>qdel</i> .                                  |
| 30996 | <i>LOGNAME</i>                | Determine the login name of the user.                                                                                |
| 30997 | <i>TZ</i>                     | Determine the timezone in which the time and date are written. If the <i>TZ</i> variable                             |
| 30998 |                               | is not set, an unspecified system default timezone is used.                                                          |
| 30999 | <b>ASYNCHRONOUS EVENTS</b>    |                                                                                                                      |
| 31000 |                               | Default.                                                                                                             |
| 31001 | <b>STDOUT</b>                 |                                                                                                                      |
| 31002 |                               | An implementation of the <i>qdel</i> utility may write informative messages to standard output.                      |
| 31003 | <b>STDERR</b>                 |                                                                                                                      |
| 31004 |                               | Used only for diagnostic messages.                                                                                   |
| 31005 | <b>OUTPUT FILES</b>           |                                                                                                                      |
| 31006 |                               | None.                                                                                                                |
| 31007 | <b>EXTENDED DESCRIPTION</b>   |                                                                                                                      |
| 31008 |                               | None.                                                                                                                |
| 31009 | <b>EXIT STATUS</b>            |                                                                                                                      |
| 31010 |                               | The following exit values shall be returned:                                                                         |
| 31011 | 0                             | Successful completion.                                                                                               |
| 31012 | >0                            | An error occurred.                                                                                                   |
| 31013 | <b>CONSEQUENCES OF ERRORS</b> |                                                                                                                      |
| 31014 |                               | In addition to the default behavior, the <i>qdel</i> utility shall not be required to write a diagnostic             |
| 31015 |                               | message to standard error when the error reply received from a batch server indicates that the                       |
| 31016 |                               | batch <i>job_identifier</i> does not exist on the server. Whether or not the <i>qdel</i> utility waits to output the |
| 31017 |                               | diagnostic message while attempting to locate the job on other servers is implementation-                            |
| 31018 |                               | dependent.                                                                                                           |
| 31019 | <b>APPLICATION USAGE</b>      |                                                                                                                      |
| 31020 |                               | None.                                                                                                                |
| 31021 | <b>EXAMPLES</b>               |                                                                                                                      |
| 31022 |                               | None.                                                                                                                |
| 31023 | <b>RATIONALE</b>              |                                                                                                                      |
| 31024 |                               | The <i>qdel</i> utility allows users and administrators to delete jobs.                                              |
| 31025 |                               | The <i>qdel</i> utility provides functionality that is not otherwise available. For example, the <i>kill</i> utility |
| 31026 |                               | of the operating system does not suffice. First, to use the <i>kill</i> utility, the user might have to log in       |
| 31027 |                               | on a remote node, because the <i>kill</i> utility does not operate across the network. Second, unlike                |
| 31028 |                               | <i>qdel</i> , <i>kill</i> cannot remove jobs from queues. Lastly, the arguments of the <i>qdel</i> utility are job   |
| 31029 |                               | identifiers rather than process identifiers, and so this utility can be passed the output of the                     |
| 31030 |                               | <i>qselect</i> utility, thus providing users with a means of deleting a list of jobs.                                |
| 31031 |                               | Because a set of jobs can be selected using the <i>qselect</i> utility, the <i>qdel</i> utility has not been         |
| 31032 |                               | complicated with options that provide for selection of jobs. Instead, the batch jobs to be deleted                   |
| 31033 |                               | are identified individually by their job identifiers.                                                                |
| 31034 |                               | Historically, the <i>qdel</i> utility has been a component of NQS, the existing practice on which it is              |
| 31035 |                               | based. However, the <i>qdel</i> utility defined in this volume of IEEE Std. 1003.1-200x does not provide             |
| 31036 |                               | an option for specifying a signal number to send to the batch job prior to the killing of the                        |
| 31037 |                               | process; that capability has been subsumed by the <i>qsig</i> utility.                                               |

31038 A discussion was held about the delays of networking and the possibility that the batch server  
31039 may never respond, due to a down router, down batch server, or other network mishap. The  
31040 DESCRIPTION records this under the words “fails to process any job identifier”. In the broad  
31041 sense, the network problem is also an error, which causes the failure to process the batch job  
31042 identifier.

31043 As with other batch utilities, implementors can extend the *qdel* utility using the **-W** option.

31044 **FUTURE DIRECTIONS**

31045 None.

31046 **SEE ALSO**

31047 *kill*, *qselect*, *qsig*, Chapter 3 on page 133

31048 **CHANGE HISTORY**

31049 Derived from IEEE Std. 1003.2d-1994.

## 31050 NAME

31051 qhold — hold batch jobs

## 31052 SYNOPSIS

31053 BE qhold [-h *hold\_list*] *job\_identifier* ...

31054

## 31055 DESCRIPTION

31056 A hold is placed on a batch job by a request to the batch server that manages the batch job. A  
 31057 batch job that has one or more holds is not eligible for execution. The *qhold* utility is a user-  
 31058 accessible client of batch services that requests one or more types of hold to be placed on one or  
 31059 more batch jobs.

31060 The *qhold* utility shall place holds on those batch jobs for which a batch *job\_identifier* is presented  
 31061 to the utility.

31062 The *qhold* utility shall place holds on batch jobs in the order in which their batch *job\_identifiers*  
 31063 are presented to the utility. If the *qhold* utility fails to process any batch *job\_identifier* successfully,  
 31064 the utility shall proceed to process the remaining batch *job\_identifiers*, if any.

31065 The *qhold* utility shall place holds on each batch job by sending a *Hold Job Request* to the batch  
 31066 server that manages the batch job.

31067 The *qhold* utility shall not exit until holds have been placed on the batch job corresponding to  
 31068 each successfully processed batch *job\_identifier*.

## 31069 OPTIONS

31070 The *qhold* utility shall conform to the System Interface Definitions volume of  
 31071 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

31072 The following option shall be supported by the implementation:

31073 **-h *hold\_list*** Define the types of holds to be placed on the batch job.

31074 The *qhold* **-h** option shall accept a value for the *hold\_list* option-argument that is a  
 31075 string of alphanumeric characters in the portable character set (see the System  
 31076 Interface Definitions volume of IEEE Std. 1003.1-200x, Section 6.1, Portable  
 31077 Character Set).

31078 The *qhold* utility shall accept a value for the *hold\_list* option-argument that is a  
 31079 string of one or more of the characters 'u', 's', or 'o', or the single character  
 31080 'n'.

31081 For each unique character in the *hold\_list* option-argument, the *qhold* utility shall  
 31082 add a value to the *Hold\_Types* attribute of the batch job as follows, each  
 31083 representing a different hold type:

31084 **u** USER

31085 **s** SYSTEM

31086 **o** OPERATOR

31087 If any of these characters are duplicated in the *hold\_list* option-argument, the  
 31088 duplicates shall be ignored.

31089 An existing *Hold\_Types* attribute can be cleared by the following hold type:

31090 **n** NO\_HOLD

31091 The *qhold* utility shall consider it an error if any hold type other than **n** is combined  
 31092 with hold type **n**.

31093 Strictly conforming applications shall not repeat any of the characters 'u', 's',  
 31094 'o', or 'n' within the *hold\_list* option-argument. The *qhold* utility shall permit the  
 31095 repetition of characters, but shall not assign additional meaning to the repeated  
 31096 characters.

31097 An implementation may define other hold types. The conformance document for  
 31098 an implementation shall describe any additional hold types, how they are  
 31099 specified, their internal behavior, and how they affect the behavior of the utility.

31100 If the *-h* option is not presented to the *qhold* utility, the implementation shall set  
 31101 the *Hold\_Types* attribute to USER.

#### 31102 OPERANDS

31103 The *qhold* utility shall accept one or more operands that conform to the syntax for a batch  
 31104 *job\_identifier* (see Section 3.3.1 on page 156).

#### 31105 STDIN

31106 Not used.

#### 31107 INPUT FILES

31108 None.

#### 31109 ENVIRONMENT VARIABLES

31110 The following environment variables shall affect the execution of *qhold*:

31111 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 31112 If *LANG* is unset or null, the corresponding value from the implementation-  
 31113 dependent default locale shall be used. If any of the internationalization variables  
 31114 contains an invalid setting, the utility shall behave as if none of the variables had  
 31115 been defined.

31116 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 31117 internationalization variables.

31118 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 31119 characters (for example, single-byte as opposed to multi-byte characters in  
 31120 arguments).

#### 31121 LC\_MESSAGES

31122 Determine the locale that should be used to affect the format and contents of  
 31123 diagnostic messages written to standard error.

31124 *LC\_TIME* Determine the format and contents of date and time strings written by *qhold*.

31125 *LOGNAME* Determine the login name of the user.

31126 *TZ* Determine the timezone in which the time and date are written. If the *TZ* variable  
 31127 is not set, an unspecified system default timezone is used.

#### 31128 ASYNCHRONOUS EVENTS

31129 Default.

#### 31130 STDOUT

31131 None.

#### 31132 STDERR

31133 Used only for diagnostic messages.

31134 **OUTPUT FILES**

31135 None.

31136 **EXTENDED DESCRIPTION**

31137 None.

31138 **EXIT STATUS**

31139 The following exit values shall be returned:

31140 0 Successful completion.

31141 &gt;0 An error occurred.

31142 **CONSEQUENCES OF ERRORS**

31143 In addition to the default behavior, the *qhold* utility shall not be required to write a diagnostic  
31144 message to standard error when the error reply received from a batch server indicates that the  
31145 batch *job\_identifier* does not exist on the server. Whether or not the *qhold* utility waits to output  
31146 the diagnostic message while attempting to locate the job on other servers is implementation-  
31147 dependent.

31148 **APPLICATION USAGE**

31149 None.

31150 **EXAMPLES**

31151 None.

31152 **RATIONALE**

31153 The *qhold* utility allows users to place a hold on one or more jobs. A hold makes a batch job  
31154 ineligible for execution.

31155 The *qhold* utility has options that allow the user to specify the type of hold. Should the user wish  
31156 to place a hold on a set of jobs that meet a selection criteria, such a list of jobs can be acquired  
31157 using the *qselect* utility.

31158 The *-h* option allows the user to specify the type of hold that is to be placed on the job. This  
31159 option allows for USER, SYSTEM, OPERATOR, and implementation-dependent hold types. The  
31160 USER and OPERATOR holds are distinct. The batch server that manages the batch job will verify  
31161 that the user is authorized to set the specified hold for the batch job.

31162 Mail is not required on hold because the administrator has the tools and libraries to build this  
31163 option if he or she wishes.

31164 As with other batch utilities, implementors may extend the *qhold* utility using the *-W* option.

31165 Historically, the *qhold* utility has been a part of some existing batch systems, although it has not  
31166 traditionally been a part of the NQS.

31167 **FUTURE DIRECTIONS**

31168 None.

31169 **SEE ALSO**31170 *qselect*, Chapter 3 on page 13331171 **CHANGE HISTORY**

31172 Derived from IEEE Std. 1003.2d-1994.

31173 **NAME**

31174           qmove — move batch jobs

31175 **SYNOPSIS**31176 BE       qmove *destination job\_identifier* ...

31177

31178 **DESCRIPTION**

31179       To move a batch job is to remove the batch job from the batch queue in which it resides and  
 31180       instantiate the batch job in another batch queue. A batch job is moved by a request to the batch  
 31181       server that manages the batch job. The *qmove* utility is a user-accessible batch client that requests  
 31182       the movement of one or more batch jobs.

31183       The *qmove* utility shall move those batch jobs, and only those batch jobs, for which a batch  
 31184       *job\_identifier* is presented to the utility.

31185       The *qmove* utility shall move batch jobs in the order in which the corresponding batch  
 31186       *job\_identifiers* are presented to the utility.

31187       If the *qmove* utility fails to process a batch *job\_identifier* successfully, the utility shall proceed to  
 31188       process the remaining batch *job\_identifiers*, if any.

31189       The *qmove* utility shall move batch jobs by sending a *Move Job Request* to the batch server that  
 31190       manages each batch job. The *qmove* utility shall not exit before the batch jobs corresponding to all  
 31191       successfully processed batch *job\_identifiers* have been moved.

31192 **OPTIONS**

31193       None.

31194 **OPERANDS**

31195       The *qmove* utility shall accept one operand that conforms to the syntax for a *destination* (see  
 31196       Section 3.3.2 on page 157).

31197       The *qmove* utility shall accept one or more operands that conform to the syntax for a batch  
 31198       *job\_identifier* (see Section 3.3.1 on page 156).

31199 **STDIN**

31200       Not used.

31201 **INPUT FILES**

31202       None.

31203 **ENVIRONMENT VARIABLES**31204       The following environment variables shall affect the execution of *qmove*:

31205       *LANG*       Provide a default value for the internationalization variables that are unset or null.  
 31206       If *LANG* is unset or null, the corresponding value from the implementation-  
 31207       dependent default locale shall be used. If any of the internationalization variables  
 31208       contains an invalid setting, the utility shall behave as if none of the variables had  
 31209       been defined.

31210       *LC\_ALL*     If set to a non-empty string value, override the values of all the other  
 31211       internationalization variables.

31212       *LC\_CTYPE*   Determine the locale for the interpretation of sequences of bytes of text data as  
 31213       characters (for example, single-byte as opposed to multi-byte characters in  
 31214       arguments).

31215       *LC\_MESSAGES*

31216       Determine the locale that should be used to affect the format and contents of

|       |                               |                                                                                                                   |
|-------|-------------------------------|-------------------------------------------------------------------------------------------------------------------|
| 31217 |                               | diagnostic messages written to standard error.                                                                    |
| 31218 | <i>LC_TIME</i>                | Determine the format and contents of date and time strings written by <i>qmove</i> .                              |
| 31219 | <i>LOGNAME</i>                | Determine the login name of the user.                                                                             |
| 31220 | <i>TZ</i>                     | Determine the timezone in which the time and date are written. If the <i>TZ</i> variable                          |
| 31221 |                               | is not set, an unspecified system default timezone is used.                                                       |
| 31222 | <b>ASYNCHRONOUS EVENTS</b>    |                                                                                                                   |
| 31223 |                               | Default.                                                                                                          |
| 31224 | <b>STDOUT</b>                 |                                                                                                                   |
| 31225 |                               | None.                                                                                                             |
| 31226 | <b>STDERR</b>                 |                                                                                                                   |
| 31227 |                               | Used only for diagnostic messages.                                                                                |
| 31228 | <b>OUTPUT FILES</b>           |                                                                                                                   |
| 31229 |                               | None.                                                                                                             |
| 31230 | <b>EXTENDED DESCRIPTION</b>   |                                                                                                                   |
| 31231 |                               | None.                                                                                                             |
| 31232 | <b>EXIT STATUS</b>            |                                                                                                                   |
| 31233 |                               | The following exit values shall be returned:                                                                      |
| 31234 | 0                             | Successful completion.                                                                                            |
| 31235 | >0                            | An error occurred.                                                                                                |
| 31236 | <b>CONSEQUENCES OF ERRORS</b> |                                                                                                                   |
| 31237 |                               | In addition to the default behavior, the <i>qmove</i> utility shall not be required to write a diagnostic         |
| 31238 |                               | message to standard error when the error reply received from a batch server indicates that the                    |
| 31239 |                               | batch <i>job_identifier</i> does not exist on the server. Whether or not the <i>qmove</i> utility waits to output |
| 31240 |                               | the diagnostic message while attempting to locate the job on other servers is implementation-                     |
| 31241 |                               | dependent.                                                                                                        |
| 31242 | <b>APPLICATION USAGE</b>      |                                                                                                                   |
| 31243 |                               | None.                                                                                                             |
| 31244 | <b>EXAMPLES</b>               |                                                                                                                   |
| 31245 |                               | None.                                                                                                             |
| 31246 | <b>RATIONALE</b>              |                                                                                                                   |
| 31247 |                               | The <i>qmove</i> utility allows users to move jobs between queues.                                                |
| 31248 |                               | The alternative to using the <i>qmove</i> utility—deleting the batch job and requeuing it—entails                 |
| 31249 |                               | considerably more typing.                                                                                         |
| 31250 |                               | Since the means of selecting jobs based on attributes has been encapsulated in the <i>qselect</i> utility,        |
| 31251 |                               | the only option of the <i>qmove</i> utility concerns authorization. The <b>-u</b> option provides the user with   |
| 31252 |                               | the convenience of changing the user identifier under which the batch job will execute.                           |
| 31253 |                               | Minimalism and consistency has taken precedence over convenience; the <b>-u</b> option has been                   |
| 31254 |                               | deleted because the equivalent capability exists with the <b>-u</b> option of the <i>qalter</i> utility.          |
| 31255 |                               | As with other batch utilities, implementors can extend the <i>qmove</i> utility using the <b>-W</b> option.       |
| 31256 |                               | The <i>qmove</i> utility is new, <i>vis-a-vis</i> existing practice; it has been defined in this volume of        |
| 31257 |                               | IEEE Std. 1003.1-200x as a logical extension of existing practice.                                                |



31258 **FUTURE DIRECTIONS**

31259 None.

31260 **SEE ALSO**31261 *qalter*, *qselect*, Chapter 3 on page 13331262 **CHANGE HISTORY**

31263 Derived from IEEE Std. 1003.2d-1994.

## 31264 NAME

31265 qmsg — send message to batch jobs

## 31266 SYNOPSIS

31267 BE qmsg [-E][-O] *message\_string* *job\_identifier* ...

31268

## 31269 DESCRIPTION

31270 To send a message to a batch job is to request that a server write a message string into one or  
 31271 more output files of the batch job. A message is sent to a batch job by a request to the batch  
 31272 server that manages the batch job. The *qmsg* utility is a user-accessible batch client that requests  
 31273 the sending of messages to one or more batch jobs.

31274 The *qmsg* utility shall write messages into the files of batch jobs by sending a *Job Message Request*  
 31275 to the batch server that manages the batch job. The *qmsg* utility shall not directly write the  
 31276 message into the files of the batch job.

31277 The *qmsg* utility shall send a *Job Message Request* for those batch jobs, and only those batch jobs,  
 31278 for which a batch *job\_identifier* is presented to the utility.

31279 The *qmsg* utility shall send *Job Message Requests* for batch jobs in the order in which their batch  
 31280 *job\_identifiers* are presented to the utility.

31281 If the *qmsg* utility fails to process any batch *job\_identifier* successfully, the utility shall proceed to  
 31282 process the remaining batch *job\_identifiers*, if any.

31283 The *qmsg* utility shall not exit before a *Job Message Request* has been sent to the server that  
 31284 manages the batch job that corresponds to each successfully processed batch *job\_identifier*.

## 31285 OPTIONS

31286 The *qmsg* utility shall conform to the System Interface Definitions volume of  
 31287 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

31288 The following options shall be supported by the implementation:

31289 -E Specify that the message is written to the standard error of each batch job.

31290 The *qmsg* utility shall write the message into the standard error of the batch job.

31291 -O Specify that the message is written to the standard output of each batch job.

31292 The *qmsg* utility shall write the message into the standard output of the batch job.

31293 If neither the -O nor the -E option is presented to the *qmsg* utility, the utility shall write the  
 31294 message into an implementation-dependent file. The conformance document for the  
 31295 implementation shall describe the name and location of the implementation-dependent file. If  
 31296 both the -O and the -E options are presented to the *qmsg* utility, then the utility shall write the  
 31297 messages to both standard output and standard error.

## 31298 OPERANDS

31299 The *qmsg* utility shall accept a minimum of two operands, *message\_string* and one or more batch  
 31300 *job\_identifiers*.

31301 The *message\_string* operand shall be the string to be written to one or more output files of the  
 31302 batch job followed by a <newline>. If the string contains <blank>s, then the string must be  
 31303 quoted. The *message\_string* shall be encoded in the portable character set (see the System  
 31304 Interface Definitions volume of IEEE Std. 1003.1-200x, Section 6.1, Portable Character Set).

31305 All remaining operands are batch *job\_identifiers* that conform to the syntax for a batch  
 31306 *job\_identifier* (see Section 3.3.1 on page 156).

31307 **STDIN**

31308 Not used.

31309 **INPUT FILES**

31310 None.

31311 **ENVIRONMENT VARIABLES**31312 The following environment variables shall affect the execution of *qmsg*:

31313 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 31314 If *LANG* is unset or null, the corresponding value from the implementation-  
 31315 dependent default locale shall be used. If any of the internationalization variables  
 31316 contains an invalid setting, the utility shall behave as if none of the variables had  
 31317 been defined.

31318 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 31319 internationalization variables.

31320 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 31321 characters (for example, single-byte as opposed to multi-byte characters in  
 31322 arguments).

31323 *LC\_MESSAGES*

31324 Determine the locale that should be used to affect the format and contents of  
 31325 diagnostic messages written to standard error.

31326 *LC\_TIME* Determine the format and contents of date and time strings written by *qmsg*.

31327 *LOGNAME* Determine the login name of the user.

31328 *TZ* Determine the timezone in which the time and date are written. If the *TZ* variable  
 31329 is not set, an unspecified system default timezone is used.

31330 **ASYNCHRONOUS EVENTS**

31331 Default.

31332 **STDOUT**

31333 None.

31334 **STDERR**

31335 Used only for diagnostic messages.

31336 **OUTPUT FILES**

31337 None.

31338 **EXTENDED DESCRIPTION**

31339 None.

31340 **EXIT STATUS**

31341 The following exit values shall be returned:

31342 0 Successful completion.

31343 &gt;0 An error occurred.

31344 **CONSEQUENCES OF ERRORS**

31345 In addition to the default behavior, the *qmsg* utility shall not be required to write a diagnostic  
 31346 message to standard error when the error reply received from a batch server indicates that the  
 31347 batch *job\_identifier* does not exist on the server. Whether or not the *qmsg* utility waits to output  
 31348 the diagnostic message while attempting to locate the job on other servers is implementation-  
 31349 dependent.

**31350 APPLICATION USAGE**

31351 None.

**31352 EXAMPLES**

31353 None.

**31354 RATIONALE**

31355 The *qmsg* utility allows users to write messages into the output files of running jobs. Users,  
31356 including operators and administrators, have a number of occasions when they want to place  
31357 messages in the output files of a batch job. For example, if a disk that is being used by a batch job  
31358 is showing errors, the operator might note this in the standard error stream of the batch job.

31359 The options of the *qmsg* utility provide users with the means of placing the message in the  
31360 output stream of their choice. The default output stream for the message—if the user does not  
31361 designate an output stream—is implementation-dependent, since many implementations will  
31362 provide, as an extension to this volume of IEEE Std. 1003.1-200x, a log file that shows the history  
31363 of utility execution.

31364 If users wish to send a message to a set of jobs that meet a selection criteria, the *qselect* utility can  
31365 be used to acquire the appropriate list of job identifiers.

31366 The **-E** option allows users to place the message in the standard error stream of the batch job.

31367 The **-O** option allows users to place the message in the standard output stream of the batch job.

31368 As with other batch utilities, implementors may extend the *qmsg* utility using to the **-W** option.

31369 Historically, the *qmsg* utility is an existing practice in the offerings of one or more implementors  
31370 of an NQS-derived batch system. The utility has been found to be useful enough that it deserves  
31371 to be included in this volume of IEEE Std. 1003.1-200x.

**31372 FUTURE DIRECTIONS**

31373 None.

**31374 SEE ALSO**

31375 *qselect*, Chapter 3 on page 133

**31376 CHANGE HISTORY**

31377 Derived from IEEE Std. 1003.2d-1994.

31378 **NAME**

31379 qrerun — rerun batch jobs

31380 **SYNOPSIS**31381 BE qrerun *job\_identifier* ...

31382

31383 **DESCRIPTION**

31384 To rerun a batch job is to terminate the session leader of the batch job, delete any associated  
 31385 checkpoint files, and return the batch job to the batch queued state. A batch job is rerun by a  
 31386 request to the batch server that manages the batch job. The *qrerun* utility is a user-accessible  
 31387 batch client that requests the rerunning of one or more batch jobs.

31388 The *qrerun* utility shall rerun those batch jobs for which a batch *job\_identifier* is presented to the  
 31389 utility.

31390 The *qrerun* utility shall rerun batch jobs in the order in which their batch *job\_identifiers* are  
 31391 presented to the utility.

31392 If the *qrerun* utility fails to process any batch *job\_identifier* successfully, the utility shall proceed  
 31393 to process the remaining batch *job\_identifiers*, if any.

31394 The *qrerun* utility shall rerun batch jobs by sending a *Rerun Job Request* to the batch server that  
 31395 manages each batch job.

31396 For each successfully processed batch *job\_identifier*, the *qrerun* utility shall have rerun the  
 31397 corresponding batch batch job at the time the utility exits.

31398 **OPTIONS**

31399 None.

31400 **OPERANDS**

31401 The *qrerun* utility shall accept one or more operands that conform to the syntax for a batch  
 31402 *job\_identifier* (see Section 3.3.1 on page 156).

31403 **STDIN**

31404 Not used.

31405 **INPUT FILES**

31406 None.

31407 **ENVIRONMENT VARIABLES**31408 The following environment variables shall affect the execution of *qrerun*:

31409 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 31410 If *LANG* is unset or null, the corresponding value from the implementation-  
 31411 dependent default locale shall be used. If any of the internationalization variables  
 31412 contains an invalid setting, the utility shall behave as if none of the variables had  
 31413 been defined.

31414 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 31415 internationalization variables.

31416 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 31417 characters (for example, single-byte as opposed to multi-byte characters in  
 31418 arguments).

31419 *LC\_MESSAGES*

31420 Determine the locale that should be used to affect the format and contents of  
 31421 diagnostic messages written to standard error.

|       |                               |                                                                                                                      |
|-------|-------------------------------|----------------------------------------------------------------------------------------------------------------------|
| 31422 | <i>LC_TIME</i>                | Determine the format and contents of date and time strings written by <i>qrerun</i> .                                |
| 31423 | <i>LOGNAME</i>                | Determine the login name of the user.                                                                                |
| 31424 | <i>TZ</i>                     | Determine the timezone in which the time and date are written. If the <i>TZ</i> variable                             |
| 31425 |                               | is not set, an unspecified system default timezone is used.                                                          |
| 31426 | <b>ASYNCHRONOUS EVENTS</b>    |                                                                                                                      |
| 31427 |                               | Default.                                                                                                             |
| 31428 | <b>STDOUT</b>                 |                                                                                                                      |
| 31429 |                               | None.                                                                                                                |
| 31430 | <b>STDERR</b>                 |                                                                                                                      |
| 31431 |                               | Used only for diagnostic messages.                                                                                   |
| 31432 | <b>OUTPUT FILES</b>           |                                                                                                                      |
| 31433 |                               | None.                                                                                                                |
| 31434 | <b>EXTENDED DESCRIPTION</b>   |                                                                                                                      |
| 31435 |                               | None.                                                                                                                |
| 31436 | <b>EXIT STATUS</b>            |                                                                                                                      |
| 31437 |                               | The following exit values shall be returned:                                                                         |
| 31438 | 0                             | Successful completion.                                                                                               |
| 31439 | >0                            | An error occurred.                                                                                                   |
| 31440 | <b>CONSEQUENCES OF ERRORS</b> |                                                                                                                      |
| 31441 |                               | In addition to the default behavior, the <i>qrerun</i> utility shall not be required to write a diagnostic           |
| 31442 |                               | message to standard error when the error reply received from a batch server indicates that the                       |
| 31443 |                               | batch <i>job_identifier</i> does not exist on the server. Whether or not the <i>qrerun</i> utility waits to output   |
| 31444 |                               | the diagnostic message while attempting to locate the job on other servers is implementation-                        |
| 31445 |                               | dependent.                                                                                                           |
| 31446 | <b>APPLICATION USAGE</b>      |                                                                                                                      |
| 31447 |                               | None.                                                                                                                |
| 31448 | <b>EXAMPLES</b>               |                                                                                                                      |
| 31449 |                               | None.                                                                                                                |
| 31450 | <b>RATIONALE</b>              |                                                                                                                      |
| 31451 |                               | The <i>qrerun</i> utility allows users to cause jobs in the running state to exit and rerun.                         |
| 31452 |                               | As with other batch utilities, implementors may extend the <i>qrerun</i> utility using the <b>-W</b> option.         |
| 31453 |                               | The <i>qrerun</i> utility is a new utility, <i>vis-a-vis</i> existing practice, that has been defined in this volume |
| 31454 |                               | of IEEE Std. 1003.1-200x to correct user-perceived deficiencies in the existing practice.                            |
| 31455 | <b>FUTURE DIRECTIONS</b>      |                                                                                                                      |
| 31456 |                               | None.                                                                                                                |
| 31457 | <b>SEE ALSO</b>               |                                                                                                                      |
| 31458 |                               | Chapter 3 on page 133                                                                                                |
| 31459 | <b>CHANGE HISTORY</b>         |                                                                                                                      |
| 31460 |                               | Derived from IEEE Std. 1003.2d-1994.                                                                                 |

## 31461 NAME

31462 qrln — release batch jobs

## 31463 SYNOPSIS

31464 BE qrln [-h *hold\_list*] *job\_identifier* ...

31465

## 31466 DESCRIPTION

31467 A batch job might have one or more holds, which prevent the batch job from executing. A batch  
 31468 job from which all the holds have been removed becomes eligible for execution and is said to  
 31469 have been released. A batch job hold is removed by sending a request to the batch server that  
 31470 manages the batch job. The *qrln* utility is a user-accessible client of batch services that requests  
 31471 holds be removed from one or more batch jobs.

31472 The *qrln* utility shall remove one or more holds from those batch jobs for which a batch  
 31473 *job\_identifier* is presented to the utility.

31474 The *qrln* utility shall remove holds from batch jobs in the order in which their batch *job\_identifiers*  
 31475 are presented to the utility.

31476 If the *qrln* utility fails to process a batch *job\_identifier* successfully, the utility shall proceed to  
 31477 process the remaining batch *job\_identifiers*, if any.

31478 The *qrln* utility shall remove holds on each batch job by sending a *Release Job Request* to the batch  
 31479 server that manages the batch job.

31480 The *qrln* utility shall not exit until the holds have been removed from the batch job  
 31481 corresponding to each successfully processed batch *job\_identifier*.

## 31482 OPTIONS

31483 The *qrln* utility shall conform to the System Interface Definitions volume of  
 31484 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

31485 The following option shall be supported by the implementation:

31486 **-h *hold\_list*** Define the types of holds to be removed from the batch job.

31487 The *qrln* **-h** option shall accept a value for the *hold\_list* option-argument that is a  
 31488 string of alphanumeric characters in the portable character set (see the System  
 31489 Interface Definitions volume of IEEE Std. 1003.1-200x, Section 6.1, Portable  
 31490 Character Set).

31491 The *qrln* utility shall accept a value for the *hold\_list* option-argument that is a string  
 31492 of one or more of the characters 'u', 's', or 'o', or the single character 'n'.

31493 For each unique character in the *hold\_list* option-argument, the *qrln* utility shall add  
 31494 a value to the *Hold\_Types* attribute of the batch job as follows, each representing a  
 31495 different hold type:

31496 **u** USER

31497 **s** SYSTEM

31498 **o** OPERATOR

31499 If any of these characters are duplicated in the *hold\_list* option-argument, the  
 31500 duplicates shall be ignored.

31501 An existing *Hold\_Types* attribute can be cleared by the following hold type:

31502 **n** NO\_HOLD

|       |                              |                                                                                                      |
|-------|------------------------------|------------------------------------------------------------------------------------------------------|
| 31503 |                              | The <i>qrls</i> utility shall consider it an error if any hold type other than <b>n</b> is combined  |
| 31504 |                              | with hold type <b>n</b> .                                                                            |
| 31505 |                              | Strictly conforming applications shall not repeat any of the characters 'u', 's',                    |
| 31506 |                              | 'o', or 'n' within the <i>hold_list</i> option-argument. The <i>qrls</i> utility shall permit the    |
| 31507 |                              | repetition of characters, but shall not assign additional meaning to the repeated                    |
| 31508 |                              | characters.                                                                                          |
| 31509 |                              | An implementation may define other hold types. The conformance document for                          |
| 31510 |                              | an implementation shall describe any additional hold types, how they are                             |
| 31511 |                              | specified, their internal behavior, and how they affect the behavior of the utility.                 |
| 31512 |                              | If the <b>-h</b> option is not presented to the <i>qrls</i> utility, the implementation shall remove |
| 31513 |                              | the <b>USER</b> hold in the <i>Hold_Types</i> attribute.                                             |
| 31514 | <b>OPERANDS</b>              |                                                                                                      |
| 31515 |                              | The <i>qrls</i> utility shall accept one or more operands that conform to the syntax for a batch     |
| 31516 |                              | <i>job_identifier</i> (see Section 3.3.1 on page 156).                                               |
| 31517 | <b>STDIN</b>                 |                                                                                                      |
| 31518 |                              | Not used.                                                                                            |
| 31519 | <b>INPUT FILES</b>           |                                                                                                      |
| 31520 |                              | None.                                                                                                |
| 31521 | <b>ENVIRONMENT VARIABLES</b> |                                                                                                      |
| 31522 |                              | The following environment variables shall affect the execution of <i>qrls</i> :                      |
| 31523 | <i>LANG</i>                  | Provide a default value for the internationalization variables that are unset or null.               |
| 31524 |                              | If <i>LANG</i> is unset or null, the corresponding value from the implementation-                    |
| 31525 |                              | dependent default locale shall be used. If any of the internationalization variables                 |
| 31526 |                              | contains an invalid setting, the utility shall behave as if none of the variables had                |
| 31527 |                              | been defined.                                                                                        |
| 31528 | <i>LC_ALL</i>                | If set to a non-empty string value, override the values of all the other                             |
| 31529 |                              | internationalization variables.                                                                      |
| 31530 | <i>LC_CTYPE</i>              | Determine the locale for the interpretation of sequences of bytes of text data as                    |
| 31531 |                              | characters (for example, single-byte as opposed to multi-byte characters in                          |
| 31532 |                              | arguments).                                                                                          |
| 31533 | <i>LC_MESSAGES</i>           |                                                                                                      |
| 31534 |                              | Determine the locale that should be used to affect the format and contents of                        |
| 31535 |                              | diagnostic messages written to standard error.                                                       |
| 31536 | <i>LC_TIME</i>               | Determine the format and contents of date and time strings written by <i>qrls</i> .                  |
| 31537 | <i>LOGNAME</i>               | Determine the login name of the user.                                                                |
| 31538 | <i>TZ</i>                    | Determine the timezone in which the time and date are written. If the <i>TZ</i> variable             |
| 31539 |                              | is not set, an unspecified system default timezone is used.                                          |
| 31540 | <b>ASYNCHRONOUS EVENTS</b>   |                                                                                                      |
| 31541 |                              | Default.                                                                                             |
| 31542 | <b>STDOUT</b>                |                                                                                                      |
| 31543 |                              | None.                                                                                                |



**31544 STDERR**

31545           Used only for diagnostic messages.

**31546 OUTPUT FILES**

31547           None.

**31548 EXTENDED DESCRIPTION**

31549           None.

**31550 EXIT STATUS**

31551           The following exit values shall be returned:

31552           0   Successful completion.

31553           >0  An error occurred.

**31554 CONSEQUENCES OF ERRORS**

31555           In addition to the default behavior, the *qrIs* utility shall not be required to write a diagnostic message to standard error when the error reply received from a batch server indicates that the batch *job\_identifier* does not exist on the server. Whether or not the *qrIs* utility waits to output the diagnostic message while attempting to locate the job on other servers is implementation-dependent.

**31560 APPLICATION USAGE**

31561           None.

**31562 EXAMPLES**

31563           None.

**31564 RATIONALE**

31565           The *qrIs* utility allows users, operators, and administrators to remove holds from jobs.

31566           The *qrIs* utility does not support any job selection options or wildcard arguments. Users may acquire a list of jobs selected by attributes using the *qselect* utility. For example, a user could select all of their held jobs.

31569           The **-h** option allows the user to specify the type of hold that is to be removed. This option allows for USER, SYSTEM, OPERATOR, and implementation-dependent hold types. The batch server that manages the batch job will verify whether the user is authorized to remove the specified hold for the batch job. If more than one type of hold has been placed on the batch job, a user may wish to remove only some of them.

31574           Mail is not required on release because the administrator has the tools and libraries to build this option if required.

31576           As with other batch utilities, implementors may extend the *qrIs* utility by means of the **-W** option.

31578           The *qrIs* utility is a new utility *vis-a-vis* existing practice; it has been defined in this volume of IEEE Std. 1003.1-200x as the natural complement to the *qhold* utility.

**31580 FUTURE DIRECTIONS**

31581           None.

**31582 SEE ALSO**

31583           *qhold*, *qselect*, Chapter 3 on page 133

|       |                                      |
|-------|--------------------------------------|
| 31584 | <b>CHANGE HISTORY</b>                |
| 31585 | Derived from IEEE Std. 1003.2d-1994. |

## 31586 NAME

31587 qselect — select batch jobs

## 31588 SYNOPSIS

```
31589 BE qselect [-a [op]date_time][-A account_string][-c [op]interval]
31590 [-h hold_list][-l resource_list][-N name][-p [op]priority]
31591 [-q destination][-r y|n][-s states][-u user_list]
31592
```

## 31593 DESCRIPTION

31594 To select a set of batch jobs is to return the batch *job\_identifiers* for each batch job that meets a list  
 31595 of selection criteria. A set of batch jobs is selected by a request to a batch server. The *qselect*  
 31596 utility is a user-accessible batch client that requests the selection of batch jobs.

31597 Upon successful completion, the *qselect* utility shall have returned a list of zero or more batch  
 31598 *job\_identifiers* that meet the criteria specified by the options and option-arguments presented to  
 31599 the utility.

31600 The *qselect* utility shall select batch jobs by sending a *Select Jobs Request* to a batch server. The  
 31601 *qselect* utility shall not exit until the server replies to each request generated.

31602 For each option presented to the *qselect* utility, the utility shall restrict the set of selected batch  
 31603 jobs as described in the OPTIONS section.

31604 The *qselect* utility shall not restrict selection of batch jobs except by authorization and as required  
 31605 by the options presented to the utility.

31606 When an option is specified with a mandatory or optional *op* component to the option-  
 31607 argument, then *op* shall specify a relation between the value of a certain batch job attribute and  
 31608 the *value* component of the option-argument. If an *op* is allowable on an option, then the  
 31609 description of the option letter indicates the *op* as either mandatory or optional. Acceptable  
 31610 strings for the *op* component, and the relation the string indicates, are shown in the following  
 31611 list:

- |       |      |                                                                                            |
|-------|------|--------------------------------------------------------------------------------------------|
| 31612 | .eq. | The value represented by the attribute of the batch job is equal to the value represented  |
| 31613 |      | by the option-argument.                                                                    |
| 31614 | .ge. | The value represented by the attribute of the batch job is greater than or equal to the    |
| 31615 |      | value represented by the option-argument.                                                  |
| 31616 | .gt. | The value represented by the attribute of the batch job is greater than the value          |
| 31617 |      | represented by the option-argument.                                                        |
| 31618 | .lt. | The value represented by the attribute of the batch job is less than the value             |
| 31619 |      | represented by the option-argument.                                                        |
| 31620 | .le. | The value represented by the attribute of the batch job is less than or equal to the value |
| 31621 |      | represented by the option-argument.                                                        |
| 31622 | .ne. | The value represented by the attribute of the batch job is not equal to the value          |
| 31623 |      | represented by the option-argument.                                                        |

## 31624 OPTIONS

31625 The *qselect* utility shall conform to the System Interface Definitions volume of  
 31626 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

31627 The following options shall be supported by the implementation:

- 31628 **-a [op]date\_time**  
 31629 Restrict selection to a specific time, or a range of times.

31630 The *qselect* utility shall select only batch jobs for which the value of the  
 31631 *Execution\_Time* attribute is related to the Epoch equivalent of the local time  
 31632 expressed by the value of the *date\_time* component of the option-argument in the  
 31633 manner indicated by the value of the *op* component of the option-argument.

31634 The *qselect* utility shall accept a *date\_time* component of the option-argument that  
 31635 conforms to the syntax of the *date\_time* operand of the *touch* utility.

31636 If the *op* component of the option-argument is not presented to the *qselect* utility,  
 31637 the utility shall select batch jobs for which the *Execution\_Time* attribute is equal to  
 31638 the *date\_time* component of the option-argument.

31639 When comparing times, the *qselect* utility shall use the following definitions for the  
 31640 *op* component of the option-argument:

31641 .eq. The time represented by value of the *Execution\_Time* attribute of the batch  
 31642 job is equal the time represented by the *date\_time* component of the  
 31643 option-argument.

31644 .ge. The time represented by value of the *Execution\_Time* attribute of the batch  
 31645 job is after or equal to the time represented by the *date\_time* component of  
 31646 the option-argument.

31647 .gt. The time represented by value of the *Execution\_Time* attribute of the batch  
 31648 job is after the time represented by the *date\_time* component of the  
 31649 option-argument.

31650 .lt. The time represented by value of the *Execution\_Time* attribute of the batch  
 31651 job is before the time represented by the *date\_time* component of the  
 31652 option-argument.

31653 .le. The time represented by value of the *Execution\_Time* attribute of the batch  
 31654 job is before or equal to the time represented by the *date\_time* component  
 31655 of the option-argument.

31656 .ne. The time represented by value of the *Execution\_Time* attribute of the batch  
 31657 job is not equal to the time represented by the *date\_time* component of the  
 31658 option-argument.

31659 The *qselect* utility shall accept the defined character strings for the *op* component of  
 31660 the option-argument.

31661 **-A *account\_string***  
 31662 Restrict selection to the batch jobs charging a specified account.

31663 The *qselect* utility shall select only batch jobs for which the value of the  
 31664 *Account\_Name* attribute of the batch job matches the value of the *account\_string*  
 31665 option-argument.

31666 The syntax of the *account\_string* option-argument is unspecified.

31667 **-c [*op*]*interval***  
 31668 Restrict selection to batch jobs within a range of checkpoint intervals.

31669 The *qselect* utility shall select only batch jobs for which the value of the *Checkpoint*  
 31670 attribute relates to the value of the *interval* component of the option-argument in  
 31671 the manner indicated by the value of the *op* component of the option-argument.

31672 If the *op* component of the option-argument is omitted, the *qselect* utility shall  
 31673 select batch jobs for which the value of the *Checkpoint* attribute is equal to the value

|       |              |                                                                                                      |  |
|-------|--------------|------------------------------------------------------------------------------------------------------|--|
| 31674 |              | of the <i>interval</i> component of the option-argument.                                             |  |
| 31675 |              | When comparing checkpoint intervals, the <i>qselect</i> utility shall use the following              |  |
| 31676 |              | definitions for the <i>op</i> component of the option-argument:                                      |  |
| 31677 |              | .eq. The value of the <i>Checkpoint</i> attribute of the batch job equals the value of               |  |
| 31678 |              | the <i>interval</i> component of the option-argument.                                                |  |
| 31679 |              | .ge. The value of the <i>Checkpoint</i> attribute of the batch job is greater than or                |  |
| 31680 |              | equal to the value of the <i>interval</i> component option-argument.                                 |  |
| 31681 |              | .gt. The value of the <i>Checkpoint</i> attribute of the batch job is greater than the               |  |
| 31682 |              | value of the <i>interval</i> component option-argument.                                              |  |
| 31683 |              | .lt. The value of the <i>Checkpoint</i> attribute of the batch job is less than the value            |  |
| 31684 |              | of the <i>interval</i> component option-argument.                                                    |  |
| 31685 |              | .le. The value of the <i>Checkpoint</i> attribute of the batch job is less than or equal             |  |
| 31686 |              | to the value of the <i>interval</i> component option-argument.                                       |  |
| 31687 |              | .ne. The value of the <i>Checkpoint</i> attribute of the batch job does not equal the                |  |
| 31688 |              | value of the <i>interval</i> component option-argument.                                              |  |
| 31689 |              | The <i>qselect</i> utility shall accept the defined character strings for the <i>op</i> component of |  |
| 31690 |              | the option-argument.                                                                                 |  |
| 31691 |              | The ordering relationship for the values of the interval option-argument is defined                  |  |
| 31692 |              | to be:                                                                                               |  |
| 31693 |              | 'n' .gt. 's' .gt. 'c=minutes' .ge. 'c'                                                               |  |
| 31694 |              | When comparing <i>Checkpoint</i> attributes with an interval having the value of the                 |  |
| 31695 |              | single character 'u', only equality or inequality are valid comparisons.                             |  |
| 31696 | -h hold_list | Restrict selection to batch jobs that have a specific type of hold.                                  |  |
| 31697 |              | The <i>qselect</i> utility shall select only batch jobs for which the value of the <i>Hold_Types</i> |  |
| 31698 |              | attribute matches the value of the <i>hold_list</i> option-argument.                                 |  |
| 31699 |              | The <i>qselect</i> -h option shall accept a value for the <i>hold_list</i> option-argument that is a |  |
| 31700 |              | string of alphanumeric characters in the portable character set (see the System                      |  |
| 31701 |              | Interface Definitions volume of IEEE Std. 1003.1-200x, Section 6.1, Portable                         |  |
| 31702 |              | Character Set).                                                                                      |  |
| 31703 |              | The <i>qselect</i> utility shall accept a value for the <i>hold_list</i> option-argument that is a   |  |
| 31704 |              | string of one or more of the characters 'u', 's', or 'o', or the single character                    |  |
| 31705 |              | 'n'.                                                                                                 |  |
| 31706 |              | Each unique character in the <i>hold_list</i> option-argument of the <i>qselect</i> utility is       |  |
| 31707 |              | defined as follows, each representing a different hold type:                                         |  |
| 31708 |              | <b>u</b> USER                                                                                        |  |
| 31709 |              | <b>s</b> SYSTEM                                                                                      |  |
| 31710 |              | <b>o</b> OPERATOR                                                                                    |  |
| 31711 |              | If any of these characters are duplicated in the <i>hold_list</i> option-argument, the               |  |
| 31712 |              | duplicates shall be ignored.                                                                         |  |
| 31713 |              | The <i>qselect</i> utility shall consider it an error if any hold type other than <b>n</b> is        |  |
| 31714 |              | combined with hold type <b>n</b> .                                                                   |  |

|       |                         |                                                                                                     |
|-------|-------------------------|-----------------------------------------------------------------------------------------------------|
| 31715 |                         | Strictly conforming applications shall not repeat any of the characters 'u', 's',                   |
| 31716 |                         | 'o', or 'n' within the <i>hold_list</i> option-argument. The <i>qselect</i> utility shall permit    |
| 31717 |                         | the repetition of characters, but shall not assign additional meaning to the repeated               |
| 31718 |                         | characters.                                                                                         |
| 31719 |                         | An implementation may define other hold types. The conformance document for                         |
| 31720 |                         | an implementation shall describe any additional hold types, how they are                            |
| 31721 |                         | specified, their internal behavior, and how they affect the behavior of the utility.                |
| 31722 | <b>-I resource_list</b> |                                                                                                     |
| 31723 |                         | Restrict selection to batch jobs with specified resource limits and attributes.                     |
| 31724 |                         | The <i>qselect</i> utility shall accept a <i>resource_list</i> option-argument with the following   |
| 31725 |                         | syntax:                                                                                             |
| 31726 |                         | <i>resource_name op value [ , , resource_name op value , , ... ]</i>                                |
| 31727 |                         | When comparing resource values, the <i>qselect</i> utility shall use the following                  |
| 31728 |                         | definitions for the <i>op</i> component of the option-argument:                                     |
| 31729 | .eq.                    | The value of the resource of the same name in the <i>Resource_List</i> attribute                    |
| 31730 |                         | of the batch job equals the value of the value component of the option-                             |
| 31731 |                         | argument.                                                                                           |
| 31732 | .ge.                    | The value of the resource of the same name in the <i>Resource_List</i> attribute                    |
| 31733 |                         | of the batch job is greater than or equal to the value of the <i>value</i>                          |
| 31734 |                         | component of the option-argument.                                                                   |
| 31735 | .gt.                    | The value of the resource of the same name in the <i>Resource_List</i> attribute                    |
| 31736 |                         | of the batch job is greater than the value of the value component of the                            |
| 31737 |                         | option-argument.                                                                                    |
| 31738 | .lt.                    | The value of the resource of the same name in the <i>Resource_List</i> attribute                    |
| 31739 |                         | of the batch job is less than the value of the value component of the                               |
| 31740 |                         | option-argument.                                                                                    |
| 31741 | .ne.                    | The value of the resource of the same name in the <i>Resource_List</i> attribute                    |
| 31742 |                         | of the batch job does not equal the value of the value component of the                             |
| 31743 |                         | option-argument.                                                                                    |
| 31744 | .le.                    | The value of the resource of the same name in the <i>Resource_List</i> attribute                    |
| 31745 |                         | of the batch job is less than or equal to the value of the <i>value</i> component                   |
| 31746 |                         | of the option-argument.                                                                             |
| 31747 |                         | When comparing the limit of a <i>Resource_List</i> attribute with the <i>value</i> component of     |
| 31748 |                         | the option-argument, if the limit, the value, or both are non-numeric, only equality                |
| 31749 |                         | or inequality are valid comparisons.                                                                |
| 31750 |                         | The <i>qselect</i> utility shall select only batch jobs for which the values of the                 |
| 31751 |                         | <i>resource_names</i> listed in the <i>resource_list</i> option-argument match the corresponding    |
| 31752 |                         | limits of the <i>Resource_List</i> attribute of the batch job.                                      |
| 31753 |                         | Limits of <i>resource_names</i> present in the <i>Resource_List</i> attribute of the batch job that |
| 31754 |                         | have no corresponding values in the <i>resource_list</i> option-argument shall not be               |
| 31755 |                         | considered when selecting batch jobs.                                                               |
| 31756 | <b>-N name</b>          | Restrict selection to batch jobs with a specified name.                                             |
| 31757 |                         | The <i>qselect</i> utility shall select only batch jobs for which the value of the <i>Job_Name</i>  |
| 31758 |                         | attribute matches the value of the <i>name</i> option-argument. The string specified in             |

|       |                        |                                                                                                          |
|-------|------------------------|----------------------------------------------------------------------------------------------------------|
| 31759 |                        | the <i>name</i> option-argument shall be passed, uninterpreted, to the server. This allows               |
| 31760 |                        | an implementation to match “wildcard” patterns against batch job names.                                  |
| 31761 |                        | An implementation shall describe in the conformance document the format it                               |
| 31762 |                        | supports for matching against the <i>Job_Name</i> attribute.                                             |
| 31763 | <b>-p [op]priority</b> |                                                                                                          |
| 31764 |                        | Restrict selection to batch jobs of the specified priority or range of priorities.                       |
| 31765 |                        | The <i>qselect</i> utility shall select only batch jobs for which the value of the <i>Priority</i>       |
| 31766 |                        | attribute of the batch job relates to the value of the <i>priority</i> component of the                  |
| 31767 |                        | option-argument in the manner indicated by the value of the <i>op</i> component of the                   |
| 31768 |                        | option-argument.                                                                                         |
| 31769 |                        | If the <i>op</i> component of the option-argument is omitted, the <i>qselect</i> utility shall           |
| 31770 |                        | select batch jobs for which the value of the <i>Priority</i> attribute of the batch job is               |
| 31771 |                        | equal to the value of the <i>priority</i> component of the option-argument.                              |
| 31772 |                        | When comparing priority values, the <i>qselect</i> utility shall use the following                       |
| 31773 |                        | definitions for the <i>op</i> component of the option-argument:                                          |
| 31774 | .eq.                   | The value of the <i>Priority</i> attribute of the batch job equals the value of the                      |
| 31775 |                        | <i>priority</i> component of the option-argument.                                                        |
| 31776 | .ge.                   | The value of the <i>Priority</i> attribute of the batch job is greater than or equal                     |
| 31777 |                        | to the value of the <i>priority</i> component option-argument.                                           |
| 31778 | .gt.                   | The value of the <i>Priority</i> attribute of the batch job is greater than the value                    |
| 31779 |                        | of the <i>priority</i> component option-argument.                                                        |
| 31780 | .lt.                   | The value of the <i>Priority</i> attribute of the batch job is less than the value of                    |
| 31781 |                        | the <i>priority</i> component option-argument.                                                           |
| 31782 | .lte.                  | The value of the <i>Priority</i> attribute of the batch job is less than or equal to                     |
| 31783 |                        | the value of the <i>priority</i> component option-argument.                                              |
| 31784 | .ne.                   | The value of the <i>Priority</i> attribute of the batch job does not equal the value                     |
| 31785 |                        | of the <i>priority</i> component option-argument.                                                        |
| 31786 | <b>-q destination</b>  |                                                                                                          |
| 31787 |                        | Restrict selection to the specified batch queue or server, or both.                                      |
| 31788 |                        | The <i>qselect</i> utility shall select only batch jobs that are located at the destination              |
| 31789 |                        | indicated by the value of the <i>destination</i> option-argument.                                        |
| 31790 |                        | The destination defines a batch queue, a server, or a batch queue at a server.                           |
| 31791 |                        | The <i>qselect</i> utility shall accept an option-argument for the <b>-q</b> option that conforms        |
| 31792 |                        | to the syntax for a destination. If the <b>-q</b> option is not presented to the <i>qselect</i> utility, |
| 31793 |                        | the utility shall select batch jobs from all batch queues at the default batch server.                   |
| 31794 |                        | If the option-argument describes only a batch queue, the <i>qselect</i> utility shall select             |
| 31795 |                        | only batch jobs from the batch queue of the specified name at the default batch                          |
| 31796 |                        | server. The means by which <i>qselect</i> determines the default server is                               |
| 31797 |                        | implementation-dependent.                                                                                |
| 31798 |                        | If the option-argument describes only a batch server, the <i>qselect</i> utility shall select            |
| 31799 |                        | batch jobs from all the batch queues at that batch server.                                               |
| 31800 |                        | If the option-argument describes both a batch queue and a batch server, the <i>qselect</i>               |
| 31801 |                        | utility shall select only batch jobs from the specified batch queue at the specified                     |

|       |                     |                                                                                                        |  |
|-------|---------------------|--------------------------------------------------------------------------------------------------------|--|
| 31802 |                     | server.                                                                                                |  |
| 31803 | <b>-r y n</b>       | Restrict selection to batch jobs with the specified rerunability status.                               |  |
| 31804 |                     | The <i>qselect</i> utility shall select only batch jobs for which the value of the <i>Rerunable</i>    |  |
| 31805 |                     | attribute of the batch job matches the value of the option-argument.                                   |  |
| 31806 |                     | The <i>qselect</i> utility shall accept a value for the option-argument that consists of               |  |
| 31807 |                     | either the single character 'y' or the single character 'n'. The character 'y'                         |  |
| 31808 |                     | represents the value TRUE, and the character 'n' represents the value FALSE.                           |  |
| 31809 | <b>-s states</b>    | Restrict selection to batch jobs in the specified states.                                              |  |
| 31810 |                     | The <i>qselect</i> utility shall accept an option-argument that consists of any combination            |  |
| 31811 |                     | of the characters 'e', 'q', 'r', 'w', 'h', and 't'.                                                    |  |
| 31812 |                     | Conforming applications shall not repeat any character in the option-argument.                         |  |
| 31813 |                     | The <i>qselect</i> utility shall permit the repetition of characters in the option-argument,           |  |
| 31814 |                     | but shall not assign additional meaning to repeated characters.                                        |  |
| 31815 |                     | The <i>qselect</i> utility shall interpret the characters in the <i>states</i> option-argument as      |  |
| 31816 |                     | follows:                                                                                               |  |
| 31817 |                     | e Represents the EXITING state.                                                                        |  |
| 31818 |                     | q Represents the QUEUED state.                                                                         |  |
| 31819 |                     | r Represents the RUNNING state.                                                                        |  |
| 31820 |                     | t Represents the TRANSITING state.                                                                     |  |
| 31821 |                     | h Represents the HELD state.                                                                           |  |
| 31822 |                     | w Represents the WAITING state.                                                                        |  |
| 31823 |                     | For each character in the <i>states</i> option-argument, the <i>qselect</i> utility shall select batch |  |
| 31824 |                     | jobs in the corresponding state.                                                                       |  |
| 31825 | <b>-u user_list</b> | Restrict selection to batch jobs owned by the specified user names.                                    |  |
| 31826 |                     | The <i>qselect</i> utility shall select only the batch jobs of those users specified in the            |  |
| 31827 |                     | <i>user_list</i> option-argument.                                                                      |  |
| 31828 |                     | The <i>qselect</i> utility shall accept a <i>user_list</i> option-argument that conforms to the        |  |
| 31829 |                     | following syntax:                                                                                      |  |
| 31830 |                     | <i>username</i> [@ <i>host</i> ][, , <i>username</i> [@ <i>host</i> ], , ...]                          |  |
| 31831 |                     | The <i>qselect</i> utility shall accept only one user name that is missing a corresponding             |  |
| 31832 |                     | host name. The <i>qselect</i> utility shall accept only one user name per named host.                  |  |
| 31833 | <b>OPERANDS</b>     |                                                                                                        |  |
| 31834 |                     | None.                                                                                                  |  |
| 31835 | <b>STDIN</b>        |                                                                                                        |  |
| 31836 |                     | Not used.                                                                                              |  |
| 31837 | <b>INPUT FILES</b>  |                                                                                                        |  |
| 31838 |                     | None.                                                                                                  |  |



31839 **ENVIRONMENT VARIABLES**

31840 The following environment variables shall affect the execution of *qselect*:

31841 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 31842 If *LANG* is unset or null, the corresponding value from the implementation-  
 31843 dependent default locale shall be used. If any of the internationalization variables  
 31844 contains an invalid setting, the utility shall behave as if none of the variables had  
 31845 been defined.

31846 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 31847 internationalization variables.

31848 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 31849 characters (for example, single-byte as opposed to multi-byte characters in  
 31850 arguments).

31851 *LC\_MESSAGES*

31852 Determine the locale that should be used to affect the format and contents of  
 31853 diagnostic messages written to standard error.

31854 *LC\_TIME* Determine the format and contents of date and time strings written by *qselect*.

31855 *LOGNAME* Determine the login name of the user.

31856 *TZ* Determine the timezone in which the time and date are written. If the *TZ* variable  
 31857 is not set, an unspecified system default timezone is used.

31858 **ASYNCHRONOUS EVENTS**

31859 Default.

31860 **STDOUT**

31861 The *qselect* utility shall write zero or more batch *job\_identifiers* to standard output.

31862 The *qselect* utility shall separate the batch *job\_identifiers* written to standard output by white  
 31863 space.

31864 The *qselect* utility shall write batch *job\_identifiers* in the following format:

31865 *sequence\_number.server\_name@server*

31866 **STDERR**

31867 Used only for diagnostic messages.

31868 **OUTPUT FILES**

31869 None.

31870 **EXTENDED DESCRIPTION**

31871 None.

31872 **EXIT STATUS**

31873 The following exit values shall be returned:

31874 0 Successful completion.

31875 >0 An error occurred.

31876 **CONSEQUENCES OF ERRORS**

31877 Default.

## 31878 APPLICATION USAGE

31879 None.

## 31880 EXAMPLES

31881 The following example shows how a user might use the *qselect* utility in conjunction with the  
 31882 *qdel* utility to delete all of his or her jobs in the queued state without affecting any jobs that are  
 31883 already running:

31884 `qdel qselect -s q`

31885 or:

31886 `qselect -s q || xargs qdel`

## 31887 RATIONALE

31888 The *qselect* utility allows users to acquire a list of job identifiers that match user-specified  
 31889 selection criteria. The list of identifiers returned by the *qselect* utility conforms to the syntax of  
 31890 the batch job identifier list processed by a utility such as *qmove*, *qdel*, and *qrls*. The *qselect* utility is  
 31891 thus a powerful tool for causing another batch system utility to act upon a set of jobs that match  
 31892 a list of selection criteria.

31893 The options of the *qselect* utility let the user apply a number of useful filters for selecting jobs.  
 31894 Each option further restricts the selection of jobs. Many of the selection options allow the  
 31895 specification of a relational operator. The FORTRAN-like syntax of the operator—that is,  
 31896 ".lt.", was chosen rather than the C-like "<=" meta-characters.

31897 The *-a* option allows users to restrict the selected jobs to those that have been submitted (or  
 31898 altered) to wait until a particular time. The time period is determined by the argument of this  
 31899 option, which includes both a time and an operator—it is thus possible to select jobs waiting  
 31900 until a specific time, jobs waiting until after a certain time, or those waiting for a time before the  
 31901 specified time.

31902 The *-A* option allows users to restrict the selected jobs to those that have been submitted (or  
 31903 altered) to charge a particular account.

31904 The *-c* option allows users to restrict the selected jobs to those whose checkpointing interval  
 31905 falls within the specified range.

31906 The *-l* option allows users to select those jobs whose resource limits fall within the range  
 31907 indicated by the value of the option. For example, a user could select those jobs for which the  
 31908 CPU time limit is greater than two hours.

31909 The *-N* option allows users to select jobs by job name. For instance, all the parts of a task that  
 31910 have been divided in parallel jobs might be given the same name, and thus manipulated as a  
 31911 group by means of this option.

31912 The *-q* option allows users to select jobs in a specified queue.

31913 The *-r* option allows users to select only those jobs with a specified rerun criteria. For instance, a  
 31914 user might select only those jobs that can be rerun for use with the *qrerun* utility.

31915 The *-s* option allows users to select only those jobs that are in a certain state.

31916 The *-u* option allows users to select jobs that have been submitted to execute under a particular  
 31917 account.

31918 As with other batch utilities, implementors can extend the *qselect* utility using the *-W* option.

31919 The selection criteria provided by the options of the *qselect* utility allow users to select jobs based  
 31920 on all the appropriate attributes that can be assigned to jobs by the *qsub* utility. When  
 31921 implementors extend the *qsub* utility, or another utilities, using the *-W* option, they may likewise

- 31922 elect to extend the *qselect* utility to allow additional selection criteria.
- 31923 Historically, the *qselect* utility has not been a part of existing practice; it is an improvement that
- 31924 has been introduced in this volume of IEEE Std. 1003.1-200x.
- 31925 **FUTURE DIRECTIONS**
- 31926 None.
- 31927 **SEE ALSO**
- 31928 *qdel*, *qrerun*, *qrls*, *qselect*, *qsub*, *touch*, Chapter 3 on page 133
- 31929 **CHANGE HISTORY**
- 31930 Derived from IEEE Std. 1003.2d-1994.

31931 **NAME**

31932           qsig — signal batch jobs

31933 **SYNOPSIS**

31934 BE       qsig [-s *signal*] *job\_identifier* ...

31935

31936 **DESCRIPTION**

31937       To signal a batch job is to send a signal to the session leader of the batch job. A batch job is  
31938       signaled by sending a request to the batch server that manages the batch job. The *qsig* utility is a  
31939       user-accessible batch client that requests the signaling of a batch job.

31940       The *qsig* utility shall signal those batch jobs for which a batch *job\_identifier* is presented to the  
31941       utility. The *qsig* utility shall not signal any batch jobs whose batch *job\_identifiers* are not  
31942       presented to the utility.

31943       The *qsig* utility shall signal batch jobs in the order in which the corresponding batch  
31944       *job\_identifiers* are presented to the utility. If the *qsig* utility fails to process a batch *job\_identifier*  
31945       successfully, the utility shall proceed to process the remaining batch *job\_identifiers*, if any.

31946       The *qsig* utility shall signal batch jobs by sending a *Signal Job Request* to the batch server that  
31947       manages the batch job.

31948       For each successfully processed batch *job\_identifier*, the *qsig* utility shall have received a  
31949       completion reply to each *Signal Job Request* sent to a batch server at the time the utility exits.

31950 **OPTIONS**

31951       The *qsig* utility shall conform to the System Interface Definitions volume of  
31952       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

31953       The following option shall be supported by the implementation:

31954       -s *signal*     Define the signal to be sent to the batch job.

31955       The *qsig* utility shall accept a *signal* option-argument that is either a symbolic  
31956       signal name or an unsigned integer signal number (see the POSIX.1-1990 standard,  
31957       Section 3.3.1.1). The *qsig* utility shall accept signal names for which the SIG prefix  
31958       has been omitted.

31959       If the *signal* option-argument is a signal name, the *qsig* utility shall send that name.

31960       If the *signal* option-argument is a number, the *qsig* utility shall send the signal  
31961       value represented by the number.

31962       If the -s option is not presented to the *qsig* utility, the utility shall send the signal  
31963       SIGTERM to each signaled batch job.

31964 **OPERANDS**

31965       The *qsig* utility shall accept one or more operands that conform to the syntax for a batch  
31966       *job\_identifier* (see Section 3.3.1 on page 156).

31967 **STDIN**

31968       Not used.

31969 **INPUT FILES**

31970       None.

31971 **ENVIRONMENT VARIABLES**

31972 The following environment variables shall affect the execution of *qsig*:

31973 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 31974 If *LANG* is unset or null, the corresponding value from the implementation-  
 31975 dependent default locale shall be used. If any of the internationalization variables  
 31976 contains an invalid setting, the utility shall behave as if none of the variables had  
 31977 been defined.

31978 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 31979 internationalization variables.

31980 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 31981 characters (for example, single-byte as opposed to multi-byte characters in  
 31982 arguments).

31983 *LC\_MESSAGES*  
 31984 Determine the locale that should be used to affect the format and contents of  
 31985 diagnostic messages written to standard error.

31986 *LC\_TIME* Determine the format and contents of date and time strings written by *qsig*.

31987 *LOGNAME* Determine the login name of the user.

31988 *TZ* Determine the timezone in which the time and date are written. If the *TZ* variable  
 31989 is not set, an unspecified system default timezone is used.

31990 **ASYNCHRONOUS EVENTS**

31991 Default.

31992 **STDOUT**

31993 An implementation of the *qsig* utility may write informative messages to standard output.

31994 **STDERR**

31995 Used only for diagnostic messages.

31996 **OUTPUT FILES**

31997 None.

31998 **EXTENDED DESCRIPTION**

31999 None.

32000 **EXIT STATUS**

32001 The following exit values shall be returned:

32002 0 Successful completion.

32003 >0 An error occurred.

32004 **CONSEQUENCES OF ERRORS**

32005 In addition to the default behavior, the *qsig* utility shall not be required to write a diagnostic  
 32006 message to standard error when the error reply received from a batch server indicates that the  
 32007 batch *job\_identifier* does not exist on the server. Whether or not the *qsig* utility waits to output the  
 32008 diagnostic message while attempting to locate the batch job on other servers is implementation-  
 32009 dependent.

32010 **APPLICATION USAGE**

32011 None.

32012 **EXAMPLES**

32013 None.

32014 **RATIONALE**32015 The *qsig* utility allows users to signal batch jobs.

32016 A user may be unable to signal a batch job with the *kill* utility of the operating system for a  
32017 number of reasons. First, the process ID of the batch job may be unknown to the user. Second,  
32018 the processes of the batch job may be on a remote node. However, by virtue of communication  
32019 between batch nodes, the *qsig* utility can arrange for the signaling of a process.

32020 Because a batch job that is not running cannot be signaled, and because the signal may not  
32021 terminate the batch job, the *qsig* utility is not a substitute for the *qdel* utility.

32022 The options of the *qsig* utility allow the user to specify the signal that is to be sent to the batch  
32023 job.

32024 The *-s* option allows users to specify a signal by name or by number, and thus override the  
32025 default signal. The POSIX.1-1990 standard defines signals by both name and number.

32026 As with other batch utilities, implementors can extend the *qsig* utility using the *-W* option.

32027 The *qsig* utility is a new utility, *vis-a-vis* existing practice; it has been defined in this volume of  
32028 IEEE Std. 1003.1-200x in response to user-perceived shortcomings in existing practice.

32029 **FUTURE DIRECTIONS**

32030 None.

32031 **SEE ALSO**32032 *kill*, *qdel*, Chapter 3 on page 13332033 **CHANGE HISTORY**

32034 Derived from IEEE Std. 1003.2d-1994.

32035 **NAME**

32036 qstat — show status of batch jobs

32037 **SYNOPSIS**32038 BE qstat [-f] *job\_identifier* ...32039 qstat -Q [-f] *destination* ...32040 qstat -B [-f] *server\_name* ...

32041

32042 **DESCRIPTION**

32043 The status of a batch job, batch queue, or batch server is obtained by a request to the server. The  
 32044 *qstat* utility is a user-accessible batch client that requests the status of one or more batch jobs,  
 32045 batch queues, or servers, and writes the status information to standard output.

32046 For each successfully processed batch *job\_identifier*, the *qstat* utility shall display information  
 32047 about the corresponding batch job.

32048 For each successfully processed destination, the *qstat* utility shall display information about the  
 32049 corresponding batch queue.

32050 For each successfully processed server name, the *qstat* utility shall display information about the  
 32051 corresponding server.

32052 The *qstat* utility shall acquire batch job status information by sending a *Job Status Request* to a  
 32053 batch server. The *qstat* utility shall acquire batch queue status information by sending a *Queue*  
 32054 *Status Request* to a batch server. The *qstat* utility shall acquire server status information by  
 32055 sending a *Server Status Request* to a batch server.

32056 **OPTIONS**

32057 The *qstat* utility shall conform to the System Interface Definitions volume of  
 32058 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

32059 The following options shall be supported by the implementation:

32060 **-f** Specify that a full display is produced.

32061 The minimum contents of a full display are specified in the STDOUT section.

32062 Additional contents and format of a full display are implementation-dependent.

32063 **-Q** Specify that the operand is a destination.

32064 The *qstat* utility shall display information about each batch queue at each  
 32065 destination identified as an operand.

32066 **-B** Specify that the operand is a server name.

32067 The *qstat* utility shall display information about each server identified as an  
 32068 operand.

32069 **OPERANDS**

32070 If the **-Q** option is presented to the *qstat* utility, the utility shall accept one or more operands that  
 32071 conform to the syntax for a destination (see Section 3.3.2 on page 157).

32072 If the **-B** option is presented to the *qstat* utility, the utility shall accept one or more *server\_name*  
 32073 operands.

32074 If neither the **-B** nor the **-Q** option is presented to the *qstat* utility, the utility shall accept one or  
 32075 more operands that conform to the syntax for a batch *job\_identifier* (see Section 3.3.1 on page  
 32076 156).

32077 **STDIN**

32078 Not used.

32079 **INPUT FILES**

32080 None.

32081 **ENVIRONMENT VARIABLES**32082 The following environment variables shall affect the execution of *qstat*:

32083 *COLUMNS* Override the system-selected horizontal screen size. See the System Interface  
 32084 Definitions volume of IEEE Std. 1003.1-200x, Chapter 8, Environment Variables for  
 32085 valid values and results when it is unset or null.

32086 *HOME* Determine the path name of the user's home directory.

32087 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 32088 If *LANG* is unset or null, the corresponding value from the implementation-  
 32089 dependent default locale shall be used. If any of the internationalization variables  
 32090 contains an invalid setting, the utility shall behave as if none of the variables had  
 32091 been defined.

32092 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 32093 internationalization variables.

32094 *LC\_COLLATE*

32095 Determine the locale for the behavior of ranges, equivalence classes and multi-  
 32096 character collating elements within regular expressions.

32097 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 32098 characters (for example, single-byte as opposed to multi-byte characters in  
 32099 arguments).

32100 *LC\_MESSAGES*

32101 Determine the locale that should be used to affect the format and contents of  
 32102 diagnostic messages written to standard error.

32103 *LC\_NUMERIC*

32104 Determine the locale for selecting the radix character used when writing floating-  
 32105 point formatted output.

32106 *LC\_TIME* Determine the format and contents of date and time strings written by *qstat*.

32107 *LINES* Override the system-selected vertical screen size, used as the number of lines in a  
 32108 screenful and the vertical screen size in visual mode. See the System Interface  
 32109 Definitions volume of IEEE Std. 1003.1-200x, Chapter 8, Environment Variables for  
 32110 valid values and results when it is unset or null.

32111 *LOGNAME* Determine the login name of the user.

32112 *TERM* Determine the terminal type. If this variable is unset or null, and if the *-T* option is  
 32113 not specified, an unspecified default terminal type shall be used.

32114 *TZ* Determine the timezone in which the time and date are written. If the *TZ* variable  
 32115 is not set, an unspecified system default timezone is used.

32116 **ASYNCHRONOUS EVENTS**

32117 Default.



32118 **STDOUT**

32119 If an operand presented to the *qstat* utility is a batch *job\_identifier* and the *-f* option is not  
 32120 specified, the *qstat* utility shall display the following items on a single line, in the stated order,  
 32121 with white space between each item, for each successfully processed operand:

- 32122 • The batch *job\_identifier*
- 32123 • The batch job name
- 32124 • The *Job\_Owner* attribute
- 32125 • The CPU time used by the batch job
- 32126 • The batch job state
- 32127 • The batch job location

32128 If an operand presented to the *qstat* utility is a batch *job\_identifier* and the *-f* option is specified,  
 32129 the *qstat* utility shall display the following items for each success fully processed operand:

- 32130 • The batch *job\_identifier*
- 32131 • The batch job name
- 32132 • The *Job\_Owner* attribute
- 32133 • The execution user ID
- 32134 • The CPU time used by the batch job
- 32135 • The batch job state
- 32136 • The batch job location
- 32137 • Additional implementation-dependent information, if any, about the batch job or batch  
 32138 queue

32139 If an operand presented to the *qstat* utility is a destination, the *-Q* option is specified, and the *-f*  
 32140 option is not specified, the *qstat* utility shall display the following items on a single line, in the  
 32141 stated order, with white space between each item, for each successfully processed operand:

- 32142 • The batch queue name
- 32143 • The maximum number of batch jobs that are allowed to run in the batch queue concurrently
- 32144 • The total number of batch jobs in the batch queue
- 32145 • The status of the batch queue
- 32146 • For each state, the number of batch jobs in that state in the batch queue and the name of the  
 32147 state
- 32148 • The type of batch queue (execution or routing)

32149 If the operands presented to the *qstat* utility are destinations, the *-Q* option is specified, and the  
 32150 *-f* option is specified, the *qstat* utility shall display the following items for each successfully  
 32151 processed operand:

- 32152 • The batch queue name
- 32153 • The maximum number of batch jobs that are allowed to run in the batch queue concurrently
- 32154 • The total number of batch jobs in the batch queue
- 32155 • The status of the batch queue

|       |                                                                                                                    |  |
|-------|--------------------------------------------------------------------------------------------------------------------|--|
| 32156 | • For each state, the number of batch jobs in that state in the batch queue and the name of the                    |  |
| 32157 | state                                                                                                              |  |
| 32158 | • The type of batch queue (execution or routing)                                                                   |  |
| 32159 | • Additional implementation-dependent information, if any, about the batch queue                                   |  |
| 32160 | If the operands presented to the <i>qstat</i> utility are batch server names, the <b>-B</b> option is specified,   |  |
| 32161 | and the <b>-f</b> option is not specified, the <i>qstat</i> utility shall display the following items on a single  |  |
| 32162 | line, in the stated order, with white space between each item, for each successfully processed                     |  |
| 32163 | operand:                                                                                                           |  |
| 32164 | • The batch server name                                                                                            |  |
| 32165 | • The maximum number of batch jobs that are allowed to run in the batch queue concurrently                         |  |
| 32166 | • The total number of batch jobs managed by the batch server                                                       |  |
| 32167 | • The status of the batch server                                                                                   |  |
| 32168 | • For each state, the number of batch jobs in that state and the name of the state                                 |  |
| 32169 | If the operands presented to the <i>qstat</i> utility are server names, the <b>-B</b> option is specified, and the |  |
| 32170 | <b>-f</b> option is specified, the <i>qstat</i> utility shall display the following items for each successfully    |  |
| 32171 | processed operand:                                                                                                 |  |
| 32172 | • The server name                                                                                                  |  |
| 32173 | • The maximum number of batch jobs that are allowed to run in the batch queue concurrently                         |  |
| 32174 | • The total number of batch jobs managed by the server                                                             |  |
| 32175 | • The status of the server                                                                                         |  |
| 32176 | • For each state, the number of batch jobs in that state and the name of the state                                 |  |
| 32177 | • Additional implementation-dependent information, if any, about the server                                        |  |
| 32178 | <b>STDERR</b>                                                                                                      |  |
| 32179 | Used only for diagnostic messages.                                                                                 |  |
| 32180 | <b>OUTPUT FILES</b>                                                                                                |  |
| 32181 | None.                                                                                                              |  |
| 32182 | <b>EXTENDED DESCRIPTION</b>                                                                                        |  |
| 32183 | None.                                                                                                              |  |
| 32184 | <b>EXIT STATUS</b>                                                                                                 |  |
| 32185 | The following exit values shall be returned:                                                                       |  |
| 32186 | 0 Successful completion.                                                                                           |  |
| 32187 | >0 An error occurred.                                                                                              |  |
| 32188 | <b>CONSEQUENCES OF ERRORS</b>                                                                                      |  |
| 32189 | In addition to the default behavior, the <i>qstat</i> utility shall not be required to write a diagnostic          |  |
| 32190 | message to standard error when the error reply received from a batch server indicates that the                     |  |
| 32191 | batch <i>job_identifier</i> does not exist on the server. Whether or not the <i>qstat</i> utility waits to output  |  |
| 32192 | the diagnostic message while attempting to locate the batch job on other servers is                                |  |
| 32193 | implementation-dependent.                                                                                          |  |

**32194 APPLICATION USAGE**

32195 None.

**32196 EXAMPLES**

32197 None.

**32198 RATIONALE**

32199 The *qstat* utility allows users to display the status of jobs and listing the batch jobs in queues.

32200 The operands of the *qstat* utility may be either job identifiers, queues (specified as destination  
32201 identifiers), or batch server names. The **-Q** and **-B** options, or absence thereof, indicate the  
32202 nature of the operands.

32203 The other options of the *qstat* utility allow the user to control the amount of information  
32204 displayed and the format in which it is displayed. Should a user wish to display the status of a  
32205 set of jobs that match a selection criteria, the *qselect* utility may be used to acquire such a list.

32206 The **-f** option allows users to request a “full” display in an implementation-dependent format.

32207 As with other batch utilities, implementors may extend the *qstat* utility using the **-W** option.

32208 Historically, the *qstat* utility has been a part of the NQS and its derivatives, the existing practice  
32209 on which it is based.

**32210 FUTURE DIRECTIONS**

32211 None.

**32212 SEE ALSO**

32213 *qselect*, Chapter 3 on page 133

**32214 CHANGE HISTORY**

32215 Derived from IEEE Std. 1003.2d-1994.

## 32216 NAME

32217            qsub — submit a script

## 32218 SYNOPSIS

```
32219 BE qsub [-a date_time][-A account_string][-c interval]
32220 [-C directive_prefix][-e path_name][-h][-j join_list][-k keep_list]
32221 [-l resource_list][-m mail_options][-M mail_list][-N name]
32222 [-o path_name][-p priority][-q destination][-r y|n]
32223 [-S path_name_list][-u user_list][-v variable_list][-V]
32224 [-z][script]
32225
```

### 32226 DESCRIPTION

To submit a script is to create a batch job that executes the script. A script is submitted by a request to a batch server. The *qsub* utility is a user-accessible batch client that submits a script.

32229 Upon successful completion, the *qsub* utility shall have created a batch job that will execute the  
32230 submitted script.

32231 The *qsub* utility shall submit a script by sending a *Queue Job Request* to a batch server.

32232 The *qsub* utility shall place the value of the following environment variables in the *Variable\_List*  
32233 attribute of the batch job: *HOME*, *LANG*, *LOGNAME*, *PATH*, *MAIL*, *SHELL*, and *TZ*. The name  
32234 of the environment variable shall be the current name prefixed with the string *PBS\_O\_*.

32235       **Note:**     If the current value of the *HOME* variable in the environment space of the *qsub* utility  
32236                   is */aa/bb/cc*, then *qsub* shall place *PBS\_O\_HOME=/aa/bb/cc* in the *Variable\_List*  
32237                   attribute of the batch job.

32238 In addition to the variables described above, the *qsub* utility shall add the following variables  
32239 with the indicated values to the variable list:

32240 *PBS\_O\_WORKDIR* The absolute path of the current working directory of the *qsub* utility process.

|       |                         |                                                                   |
|-------|-------------------------|-------------------------------------------------------------------|
| 32241 | <code>PBS_O_HOST</code> | The name of the host on which the <i>qsub</i> utility is running. |
|-------|-------------------------|-------------------------------------------------------------------|

## 32242 OPTIONS

32243 The *qsub* utility shall conform to the System Interface Definitions volume of  
32244 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

32245 The following options shall be supported by the implementation:

32246        **-a *date\_time*** Define the time at which a batch job becomes eligible for execution.

32247 The *qsub* utility shall accept an option-argument that conforms to the syntax of the  
32248 *date\_time* operand of the *touch* utility.

32249 **Table 4-18** Environment Variable Values (Utilities)

32250  
32251  
32252  
32253  
32254  
32255  
32256  
32257  
32258  
32259  
32260

| Variable Name        | Value at qsub Time        |
|----------------------|---------------------------|
| <i>PBS_O_HOME</i>    | <i>HOME</i>               |
| <i>PBS_O_HOST</i>    | Client host name          |
| <i>PBS_O_LANG</i>    | <i>LANG</i>               |
| <i>PBS_O_LOGNAME</i> | <i>LOGNAME</i>            |
| <i>PBS_O_PATH</i>    | <i>PATH</i>               |
| <i>PBS_O_MAIL</i>    | <i>MAIL</i>               |
| <i>PBS_O_SHELL</i>   | <i>SHELL</i>              |
| <i>PBS_O_TZ</i>      | <i>TZ</i>                 |
| <i>PBS_O_WORKDIR</i> | Current working directory |

32261  
32262  
32263

**Note:** The server that initiates execution of the batch job will add other variables to the batch job's environment; see Section 3.2.2.1 on page 139.

32264  
32265  
32266  
32267

The *qsub* utility shall set the *Execution\_Time* attribute of the batch job to the number of seconds since the Epoch that is equivalent to the local time expressed by the value of the *date\_time* option-argument. The Epoch is defined in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 3.151, Epoch.

32268  
32269  
32270

If the *-a* option is not presented to the *qsub* utility, the utility shall set the *Execution\_Time* attribute of the batch job to a time (number of seconds since the Epoch) that is earlier than the time at which the utility exits.

32271  
32272  
32273

**-A *account\_string***

Define the account to which the resource consumption of the batch job should be charged.

32274

The syntax of the *account\_string* option-argument is unspecified.

32275  
32276

The *qsub* utility shall set the *Account\_Name* attribute of the batch job to the value of the *account\_string* option-argument.

32277  
32278

If the *-A* option is not presented to the *qsub* utility, the utility shall omit the *Account\_Name* attribute from the attributes of the batch job.

32279 **-c *interval***

Define whether the batch job should be checkpointed, and if so, how often.

32280  
32281

The *qsub* utility shall accept a value for the interval option-argument that is one of the following:

32282  
32283

*n* No checkpointing shall be performed on the batch batch job (NO\_CHECKPOINT).

32284  
32285

*s* Checkpointing shall be performed only when the batch server is shut down (CHECKPOINT\_AT\_SHUTDOWN).

32286  
32287  
32288

*c* Automatic periodic checkpointing shall be performed at the *Minimum\_Cpu\_Interval* attribute of the batch queue, in units of CPU minutes (CHECKPOINT\_AT\_MIN\_CPU\_INTERVAL).

32289  
32290  
32291  
32292

*c=minutes* Automatic periodic checkpointing shall be performed every minutes of CPU time, or every *Minimum\_Cpu\_Interval* minutes, whichever is greater. The *minutes* argument shall conform to the syntax for unsigned integers and shall be greater than zero.

32293  
32294

The *qsub* utility shall set the *Checkpoint* attribute of the batch job to the value of the *interval* option-argument.

|       |                                  |                                                                                                                     |
|-------|----------------------------------|---------------------------------------------------------------------------------------------------------------------|
| 32295 |                                  | If the <code>-c</code> option is not presented to the <i>qsub</i> utility, the utility shall set the                |
| 32296 |                                  | <i>Checkpoint</i> attribute of the batch job to the single character 'u'                                            |
| 32297 |                                  | (CHECKPOINT_UNSPECIFIED).                                                                                           |
| 32298 | <code>-C directive_prefix</code> |                                                                                                                     |
| 32299 |                                  | Define the prefix that declares a directive to the <i>qsub</i> utility within the script.                           |
| 32300 |                                  | The <i>directive_prefix</i> is not a batch job attribute; it affects the behavior of the <i>qsub</i>                |
| 32301 |                                  | utility.                                                                                                            |
| 32302 |                                  | If the <code>-C</code> option is presented to the <i>qsub</i> utility, and the value of the <i>directive_prefix</i> |
| 32303 |                                  | option-argument is the null string, the utility shall not scan the script file for                                  |
| 32304 |                                  | directives. If the <code>-C</code> option is not presented to the <i>qsub</i> utility, then the value of the        |
| 32305 |                                  | <i>PBS_DPREFIX</i> environment variable is used. If the environment variable is not                                 |
| 32306 |                                  | defined, then #PBS encoded in the portable character set is the default.                                            |
| 32307 | <code>-e path_name</code>        | Define the path to be used for the standard error stream of the batch job.                                          |
| 32308 |                                  | The <i>qsub</i> utility shall accept a <i>path_name</i> option-argument that conforms to the                        |
| 32309 |                                  | syntax of the <i>path_name</i> element defined in the POSIX.1-1990 standard, which can                              |
| 32310 |                                  | be preceded by a host name element of the form <i>hostname</i> :                                                    |
| 32311 |                                  | If the <i>path_name</i> option-argument constitutes an absolute path name, the <i>qsub</i>                          |
| 32312 |                                  | utility shall set the <i>Error_Path</i> attribute of the batch job to the value of the                              |
| 32313 |                                  | <i>path_name</i> option-argument.                                                                                   |
| 32314 |                                  | If the <i>path_name</i> option-argument constitutes a relative path name and no host                                |
| 32315 |                                  | name element is specified, the <i>qsub</i> utility shall set the <i>Error_Path</i> attribute of the                 |
| 32316 |                                  | batch job to the value of the absolute path name derived by expanding the                                           |
| 32317 |                                  | <i>path_name</i> option-argument relative to the current directory of the process                                   |
| 32318 |                                  | executing <i>qsub</i> .                                                                                             |
| 32319 |                                  | If the <i>path_name</i> option-argument constitutes a relative path name and a host name                            |
| 32320 |                                  | element is specified, the <i>qsub</i> utility shall set the <i>Error_Path</i> attribute of the batch                |
| 32321 |                                  | job to the value of the <i>path_name</i> option-argument without expansion. The host                                |
| 32322 |                                  | name element shall be included.                                                                                     |
| 32323 |                                  | If the <i>path_name</i> option-argument does not include a host name element, the <i>qsub</i>                       |
| 32324 |                                  | utility shall prefix the path name with <i>hostname</i> ., where <i>hostname</i> is the name of the                 |
| 32325 |                                  | host upon which the <i>qsub</i> utility is being executed.                                                          |
| 32326 |                                  | If the <code>-e</code> option is not presented to the <i>qsub</i> utility, the utility shall set the                |
| 32327 |                                  | <i>Error_Path</i> attribute of the batch job to the host name and path of the current                               |
| 32328 |                                  | directory of the submitting process and the default file name.                                                      |
| 32329 |                                  | The default file name for standard error has the following format:                                                  |
| 32330 |                                  | <i>job_name.e</i> <i>sequence_number</i>                                                                            |
| 32331 | <code>-h</code>                  | Specify that a USER hold is applied to the batch job.                                                               |
| 32332 |                                  | The <i>qsub</i> utility shall set the value of the <i>Hold_Types</i> attribute of the batch job to the              |
| 32333 |                                  | value USER.                                                                                                         |
| 32334 |                                  | If the <code>-h</code> option is not presented to the <i>qsub</i> utility, the utility shall set the                |
| 32335 |                                  | <i>Hold_Types</i> attribute of the batch job to the value NO_HOLD.                                                  |
| 32336 | <code>-j join_list</code>        | Define which streams of the batch job are to be merged. The <i>qsub -j</i> option shall                             |
| 32337 |                                  | accept a value for the <i>join_list</i> option-argument that is a string of alphanumeric                            |
| 32338 |                                  | characters in the portable character set (see the System Interface Definitions                                      |

|       |                     |                                                                                                           |
|-------|---------------------|-----------------------------------------------------------------------------------------------------------|
| 32339 |                     | volume of IEEE Std. 1003.1-200x, Section 6.1, Portable Character Set).                                    |
| 32340 |                     | The <i>qsub</i> utility shall accept a <i>join_list</i> option-argument that consists of one or           |
| 32341 |                     | more of the characters 'e' and 'o' or the single character 'n'.                                           |
| 32342 |                     | All of the other batch job output streams specified will be merged into the output                        |
| 32343 |                     | stream represented by the character listed first in the <i>join_list</i> option-argument.                 |
| 32344 |                     | For each unique character in the <i>join_list</i> option-argument, the <i>qsub</i> utility shall          |
| 32345 |                     | add a value to the <i>Join_Path</i> attribute of the batch job as follows, each representing              |
| 32346 |                     | a different batch job stream to join:                                                                     |
| 32347 | <i>e</i>            | The standard error of the batch batch job (JOIN_STD_ERROR).                                               |
| 32348 | <i>o</i>            | The standard output of the batch batch job (JOIN_STD_OUTPUT).                                             |
| 32349 |                     | An existing <i>Join_Path</i> attribute can be cleared by the following join type:                         |
| 32350 | <b>n</b>            | NO_JOIN                                                                                                   |
| 32351 |                     | If <b>n</b> is specified, then no files are joined. The <i>qsub</i> utility shall consider it an error if |
| 32352 |                     | any join type other than <b>n</b> is combined with join type <b>n</b> .                                   |
| 32353 |                     | Strictly conforming applications shall not repeat any of the characters 'e', 'o', or                      |
| 32354 |                     | 'n' within the <i>join_list</i> option-argument. The <i>qsub</i> utility shall permit the                 |
| 32355 |                     | repetition of characters, but shall not assign additional meaning to the repeated                         |
| 32356 |                     | characters.                                                                                               |
| 32357 |                     | An implementation may define other join types. The conformance document for an                            |
| 32358 |                     | implementation shall describe any additional batch job streams, how they are                              |
| 32359 |                     | specified, their internal behavior, and how they affect the behavior of the utility.                      |
| 32360 |                     | If the <b>-j</b> option is not presented to the <i>qsub</i> utility, the utility shall set the value of   |
| 32361 |                     | the <i>Join_Path</i> attribute of the batch job to NO_JOIN.                                               |
| 32362 | <b>-k keep_list</b> | Define which output of the batch job to retain on the execution host.                                     |
| 32363 |                     | The <i>qsub</i> <b>-k</b> option shall accept a value for the <i>keep_list</i> option-argument that is a  |
| 32364 |                     | string of alphanumeric characters in the portable character set (see the System                           |
| 32365 |                     | Interface Definitions volume of IEEE Std. 1003.1-200x, Section 6.1, Portable                              |
| 32366 |                     | Character Set).                                                                                           |
| 32367 |                     | The <i>qsub</i> utility shall accept a <i>keep_list</i> option-argument that consists of one or           |
| 32368 |                     | more of the characters 'e' and 'o' or the single character 'n'.                                           |
| 32369 |                     | For each unique character in the <i>keep_list</i> option-argument, the <i>qsub</i> utility shall          |
| 32370 |                     | add a value to the <i>Keep_Files</i> attribute of the batch job as follows, each representing             |
| 32371 |                     | a different batch job stream to keep:                                                                     |
| 32372 | <i>e</i>            | The standard error of the batch batch job (KEEP_STD_ERROR).                                               |
| 32373 | <i>o</i>            | The standard output of the batch batch job (KEEP_STD_OUTPUT).                                             |
| 32374 |                     | If both <i>e</i> and <i>o</i> are specified, then both files are retained. An existing <i>Keep_Files</i>  |
| 32375 |                     | attribute can be cleared by the following keep type:                                                      |
| 32376 | <b>n</b>            | NO_KEEP                                                                                                   |
| 32377 |                     | If <b>n</b> is specified, then no files are retained. The <i>qsub</i> utility shall consider it an error  |
| 32378 |                     | if any keep type other than <b>n</b> is combined with keep type <b>n</b> .                                |
| 32379 |                     | Strictly conforming applications shall not repeat any of the characters 'e', 'o', or                      |
| 32380 |                     | 'n' within the <i>keep_list</i> option-argument. The <i>qsub</i> utility shall permit the                 |

|       |                         |                                                                                                               |
|-------|-------------------------|---------------------------------------------------------------------------------------------------------------|
| 32381 |                         | repetition of characters, but shall not assign additional meaning to the repeated                             |
| 32382 |                         | characters.                                                                                                   |
| 32383 |                         | An implementation may define other keep types. The conformance document for                                   |
| 32384 |                         | an implementation shall describe any additional keep types, how they are                                      |
| 32385 |                         | specified, their internal behavior, and how they affect the behavior of the utility. If                       |
| 32386 |                         | the <b>-k</b> option is not presented to the <i>qsub</i> utility, the utility shall set the <i>Keep_Files</i> |
| 32387 |                         | attribute of the batch job to the value <b>NO_KEEP</b> .                                                      |
| 32388 | <b>-l resource_list</b> |                                                                                                               |
| 32389 |                         | Define the resources that are allowed or required by the batch job.                                           |
| 32390 |                         | The <i>qsub</i> utility shall accept a <i>resource_list</i> option-argument that conforms to the              |
| 32391 |                         | following syntax:                                                                                             |
| 32392 |                         | <i>resource=value[ , , resource=value, , ... ]</i>                                                            |
| 32393 |                         | For each resource listed in the <i>resource_list</i> option-argument, the <i>qsub</i> utility shall           |
| 32394 |                         | add one entry in the <i>Resource_List</i> attribute of the batch job, each such entry                         |
| 32395 |                         | containing the name of the resource and the value.                                                            |
| 32396 |                         | If the <b>-l</b> option is not presented to the <i>qsub</i> utility, the utility shall omit the               |
| 32397 |                         | <i>Resource_List</i> attribute from the attributes of the batch job. See Section 3.3.3 on page                |
| 32398 |                         | 157 for a means of removing <i>keyword=value</i> (and <i>value@keyword</i> ) pairs, and other                 |
| 32399 |                         | general rules for list-oriented batch job attributes.                                                         |
| 32400 | <b>Note:</b>            | See <REFERENCE UNDEFINED>(???)Table for a list of the reserved                                                |
| 32401 |                         | resource names.                                                                                               |
| 32402 | <b>-m mail_options</b>  |                                                                                                               |
| 32403 |                         | Define the points in the execution of the batch job at which the batch server that                            |
| 32404 |                         | manages the batch job shall send mail about a change in the state of the batch job.                           |
| 32405 |                         | The <i>qsub -m</i> option shall accept a value for the <i>mail_options</i> option-argument that               |
| 32406 |                         | is a string of alphanumeric characters in the portable character set (see the System                          |
| 32407 |                         | Interface Definitions volume of IEEE Std. 1003.1-200x, Section 6.1, Portable                                  |
| 32408 |                         | Character Set).                                                                                               |
| 32409 |                         | The <i>qsub</i> utility shall accept a value for the <i>mail_options</i> option-argument that is a            |
| 32410 |                         | string of one or more of the characters 'e', 'b', and 'a', or the single character                            |
| 32411 |                         | 'n'.                                                                                                          |
| 32412 |                         | For each unique character in the <i>mail_options</i> option-argument, the <i>qsub</i> utility shall           |
| 32413 |                         | add a value to the <i>Mail_Users</i> attribute of the batch job as follows, each                              |
| 32414 |                         | representing a different time during the life of a batch job at which to send mail:                           |
| 32415 | <i>e</i>                | <b>MAIL_AT_EXIT</b>                                                                                           |
| 32416 | <i>b</i>                | <b>MAIL_AT_BEGINNING</b>                                                                                      |
| 32417 | <i>a</i>                | <b>MAIL_AT_ABORT</b>                                                                                          |
| 32418 |                         | If any of these characters are duplicated in the <i>mail_options</i> option-argument, the                     |
| 32419 |                         | duplicates shall be ignored.                                                                                  |
| 32420 |                         | An existing <i>Mail_Points</i> attribute can be cleared by the following mail type:                           |
| 32421 | <b>n</b>                | <b>NO_MAIL</b>                                                                                                |
| 32422 |                         | If <b>n</b> is specified, then mail is not sent. The <i>qsub</i> utility shall consider it an error if        |
| 32423 |                         | any mail type other than <b>n</b> is combined with mail type <b>n</b> .                                       |



|       |                     |                                                                                                        |
|-------|---------------------|--------------------------------------------------------------------------------------------------------|
| 32424 |                     | Strictly conforming applications shall not repeat any of the characters 'e', 'b',                      |
| 32425 |                     | 'a', or 'n' within the <i>mail_options</i> option-argument.                                            |
| 32426 |                     | The <i>qsub</i> utility shall permit the repetition of characters, but shall not assign                |
| 32427 |                     | additional meaning to the repeated characters. An implementation may define                            |
| 32428 |                     | other mail types. The conformance document for an implementation shall describe                        |
| 32429 |                     | any additional mail types, how they are specified, their internal behavior, and how                    |
| 32430 |                     | they affect the behavior of the utility.                                                               |
| 32431 |                     | If the <i>-m</i> option is not presented to the <i>qsub</i> utility, the utility shall set the         |
| 32432 |                     | <i>Mail_Points</i> attribute to the value MAIL_AT_ABORT.                                               |
| 32433 | <i>-M mail_list</i> | Define the list of users to which a batch server that executes the batch job shall                     |
| 32434 |                     | send mail, if the server sends mail about the batch job.                                               |
| 32435 |                     | The syntax of the <i>mail_list</i> option-argument is unspecified.                                     |
| 32436 |                     | If the implementation of the <i>qsub</i> utility uses a name service to locate users, the              |
| 32437 |                     | utility should accept the syntax used by the name service.                                             |
| 32438 |                     | If the implementation of the <i>qsub</i> utility does not use a name service to locate                 |
| 32439 |                     | users, the implementation should accept the following syntax for user names:                           |
| 32440 |                     | <i>mail_address</i> [ , , <i>mail_address</i> , , ... ]                                                |
| 32441 |                     | The interpretation of <i>mail_address</i> is implementation-dependent.                                 |
| 32442 |                     | The <i>qsub</i> utility shall set the <i>Mail_Users</i> attribute of the batch job to the value of the |
| 32443 |                     | <i>mail_list</i> option-argument.                                                                      |
| 32444 |                     | If the <i>-M</i> option is not presented to the <i>qsub</i> utility, the utility shall place only the  |
| 32445 |                     | user name and host name for the current process in the <i>Mail_Users</i> attribute of the              |
| 32446 |                     | batch job.                                                                                             |
| 32447 | <i>-N name</i>      | Define the name of the batch job.                                                                      |
| 32448 |                     | The <i>qsub</i> <i>-N</i> option shall accept a value for the <i>name</i> option-argument that is a    |
| 32449 |                     | string of up to 15 alphanumeric characters in the portable character set (see the                      |
| 32450 |                     | System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 6.1, Portable                    |
| 32451 |                     | Character Set) where the first character is alphabetic.                                                |
| 32452 |                     | The <i>qsub</i> utility shall set the value of the <i>Job_Name</i> attribute of the batch job to the   |
| 32453 |                     | value of the <i>name</i> option-argument.                                                              |
| 32454 |                     | If the <i>-N</i> option is not presented to the <i>qsub</i> utility, the utility shall set the         |
| 32455 |                     | <i>Job_Name</i> attribute of the batch job to the name of the <i>script</i> argument from which        |
| 32456 |                     | the directory specification if any, has been removed.                                                  |
| 32457 |                     | If the <i>-N</i> option is not presented to the <i>qsub</i> utility, and the script is read from       |
| 32458 |                     | standard input, the utility shall set the <i>Job_Name</i> attribute of the batch job to the            |
| 32459 |                     | value STDIN.                                                                                           |
| 32460 | <i>-o path_name</i> | Define the path for the standard output of the batch job.                                              |
| 32461 |                     | The <i>qsub</i> utility shall accept a <i>path_name</i> option-argument that conforms to the           |
| 32462 |                     | syntax of the <i>path_name</i> element defined in the POSIX.1-1990 standard, which can                 |
| 32463 |                     | be preceded by a host name element of the form <i>hostname</i> :                                       |
| 32464 |                     | If the <i>path_name</i> option-argument constitutes an absolute path name, the <i>qsub</i>             |
| 32465 |                     | utility shall set the <i>Output_Path</i> attribute of the batch job to the value of the                |
| 32466 |                     | <i>path_name</i> option-argument without expansion.                                                    |

|       |                       |                                                                                                                  |
|-------|-----------------------|------------------------------------------------------------------------------------------------------------------|
| 32467 |                       | If the <i>path_name</i> option-argument constitutes a relative path name and no host                             |
| 32468 |                       | name element is specified, the <i>qsub</i> utility shall set the <i>Output_Path</i> attribute of the             |
| 32469 |                       | batch job to the path name derived by expanding the value of the <i>path_name</i>                                |
| 32470 |                       | option-argument relative to the current directory of the process executing the <i>qsub</i> .                     |
| 32471 |                       | If the <i>path_name</i> option-argument constitutes a relative path name and a host name                         |
| 32472 |                       | element is specified, the <i>qsub</i> utility shall set the <i>Output_Path</i> attribute of the batch            |
| 32473 |                       | job to the value of the <i>path_name</i> option-argument without expansion.                                      |
| 32474 |                       | If the <i>path_name</i> option-argument does not specify a host name element, the <i>qsub</i>                    |
| 32475 |                       | utility shall prefix the path name with <i>hostname</i> ., where <i>hostname</i> is the name of the              |
| 32476 |                       | host upon which the <i>qsub</i> utility is executing.                                                            |
| 32477 |                       | If the <i>-o</i> option is not presented to the <i>qsub</i> utility, the utility shall set the                   |
| 32478 |                       | <i>Output_Path</i> attribute of the batch job to the host name and path of the current                           |
| 32479 |                       | directory of the submitting process and the default file name.                                                   |
| 32480 |                       | The default file name for standard output has the following format:                                              |
| 32481 |                       | <i>job_name.osequence_number</i>                                                                                 |
| 32482 | <i>-p priority</i>    | Define the priority the batch job should have relative to other batch jobs owned by                              |
| 32483 |                       | the batch server.                                                                                                |
| 32484 |                       | The <i>qsub</i> utility shall set the <i>Priority</i> attribute of the batch job to the value of the             |
| 32485 |                       | <i>priority</i> option-argument.                                                                                 |
| 32486 |                       | If the <i>-p</i> option is not presented to the <i>qsub</i> utility, the value of the <i>Priority</i>            |
| 32487 |                       | attribute is implementation-dependent.                                                                           |
| 32488 |                       | The <i>qsub</i> utility shall accept a value for the <i>priority</i> option-argument that conforms               |
| 32489 |                       | to the syntax for signed decimal integers, and which is not less than -1 024 and not                             |
| 32490 |                       | greater than 1 023.                                                                                              |
| 32491 | <i>-q destination</i> |                                                                                                                  |
| 32492 |                       | Define the destination of the batch job.                                                                         |
| 32493 |                       | The destination is not a batch job attribute; it determines the batch server, and                                |
| 32494 |                       | possibly the batch queue, to which the <i>qsub</i> utility batch queues the batch job.                           |
| 32495 |                       | The <i>qsub</i> utility shall submit the script to the batch server named by the <i>destination</i>              |
| 32496 |                       | option-argument or the server that owns the batch queue named in the <i>destination</i>                          |
| 32497 |                       | option-argument.                                                                                                 |
| 32498 |                       | The <i>qsub</i> utility shall accept an option-argument for the <i>-q</i> option that conforms to                |
| 32499 |                       | the syntax for a destination (see Section 3.3.2 on page 157).                                                    |
| 32500 |                       | If the <i>-q</i> option is not presented to the <i>qsub</i> utility, the <i>qsub</i> utility shall submit the    |
| 32501 |                       | batch job to the default destination. The mechanism for determining the default                                  |
| 32502 |                       | destination is implementation-dependent.                                                                         |
| 32503 | <i>-r y   n</i>       | Define whether the batch job is rerunnable.                                                                      |
| 32504 |                       | If the value of the option-argument is <i>yy</i> , the <i>qsub</i> utility shall set the <i>Rerunnable</i>       |
| 32505 |                       | attribute of the batch job to TRUE.                                                                              |
| 32506 |                       | If the value of the option-argument is <i>nn</i> , the <i>qsub</i> utility shall set the <i>Rerunnable</i>       |
| 32507 |                       | attribute of the batch job to FALSE.                                                                             |
| 32508 |                       | If the <i>-r</i> option is not presented to the <i>qsub</i> utility, the utility shall set the <i>Rerunnable</i> |
| 32509 |                       | attribute of the batch job to TRUE.                                                                              |

|       |                                 |                                                                                                                 |
|-------|---------------------------------|-----------------------------------------------------------------------------------------------------------------|
| 32510 | <b>-S <i>path_name_list</i></b> |                                                                                                                 |
| 32511 |                                 | Define the path name to the shell under which the batch job is to execute.                                      |
| 32512 |                                 | The <i>qsub</i> utility shall accept a <i>path_name_list</i> option-argument that conforms to the               |
| 32513 |                                 | following syntax:                                                                                               |
| 32514 |                                 | <i>pathname</i> [@ <i>host</i> ][, , <i>pathname</i> [@ <i>host</i> ], , ...]                                   |
| 32515 |                                 | The <i>qsub</i> utility shall allow only one path name for a given host name. The <i>qsub</i>                   |
| 32516 |                                 | utility shall allow only one path name that is missing a corresponding host name.                               |
| 32517 |                                 | The <i>qsub</i> utility shall add a value to the <i>Shell_Path_List</i> attribute of the batch job for          |
| 32518 |                                 | each entry in the <i>path_name_list</i> option-argument.                                                        |
| 32519 |                                 | If the <b>-S</b> option is not presented to the <i>qsub</i> utility, the utility shall set the                  |
| 32520 |                                 | <i>Shell_Path_List</i> attribute of the batch job to the null string.                                           |
| 32521 |                                 | The conformance document for an implementation shall describe the mechanism                                     |
| 32522 |                                 | used to set the default shell and determine the current value of the default shell.                             |
| 32523 |                                 | An implementation shall provide a means for the installation to set the default                                 |
| 32524 |                                 | shell to the login shell of the user under which the batch job is to execute. See                               |
| 32525 |                                 | Section 3.3.3 on page 157 for a means of removing <i>keyword=value</i> (and                                     |
| 32526 |                                 | <i>value@keyword</i> ) pairs and other general rules for list-oriented batch job attributes.                    |
| 32527 | <b>-u <i>user_list</i></b>      | Define the user name under which the batch job is to execute.                                                   |
| 32528 |                                 | The <i>qsub</i> utility shall accept a <i>user_list</i> option-argument that conforms to the                    |
| 32529 |                                 | following syntax:                                                                                               |
| 32530 |                                 | <i>username</i> [@ <i>host</i> ][, , <i>username</i> [@ <i>host</i> ], , ...]                                   |
| 32531 |                                 | The <i>qsub</i> utility shall accept only one user name that is missing a corresponding                         |
| 32532 |                                 | host name. The <i>qsub</i> utility shall accept only one user name per named host.                              |
| 32533 |                                 | The <i>qsub</i> utility shall add a value to the <i>User_List</i> attribute of the batch job for each           |
| 32534 |                                 | entry in the <i>user_list</i> option-argument.                                                                  |
| 32535 |                                 | If the <b>-u</b> option is not presented to the <i>qsub</i> utility, the utility shall set the <i>User_List</i> |
| 32536 |                                 | attribute of the batch job to the user name from which the utility is executing. See                            |
| 32537 |                                 | Section 3.3.3 on page 157 for a means of removing <i>keyword=value</i> (and                                     |
| 32538 |                                 | <i>value@keyword</i> ) pairs and other general rules for list-oriented batch job attributes.                    |
| 32539 | <b>-v <i>variable_list</i></b>  |                                                                                                                 |
| 32540 |                                 | Add to the list of variables that are exported to the session leader of the batch job.                          |
| 32541 |                                 | A <i>variable_list</i> is a set of strings of either the form < <i>variable</i> > or < <i>variable=value</i> >, |
| 32542 |                                 | delimited by commas.                                                                                            |
| 32543 |                                 | If the <b>-v</b> option is presented to the <i>qsub</i> utility, the utility shall also add, to the             |
| 32544 |                                 | environment <i>Variable_List</i> attribute of the batch job, every variable named in the                        |
| 32545 |                                 | environment <i>variable_list</i> option-argument and, optionally, values of specified                           |
| 32546 |                                 | variables.                                                                                                      |
| 32547 |                                 | If a value is not provided on the command line, the <i>qsub</i> utility shall set the value                     |
| 32548 |                                 | of each variable in the environment <i>Variable_List</i> attribute of the batch job to the                      |
| 32549 |                                 | value of the corresponding environment variable for the process in which the                                    |
| 32550 |                                 | utility is executing; see Table 4-18 on page 840.                                                               |
| 32551 |                                 | A conforming application shall not repeat a variable in the environment                                         |
| 32552 |                                 | <i>variable_list</i> option-argument.                                                                           |

|       |                              |                                                                                                                                                                                                                                                                                                                                                                  |
|-------|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 32553 |                              | The <i>qsub</i> utility shall not repeat a variable in the environment <i>Variable_List</i> attribute of the batch job. See Section 3.3.3 on page 157 for a means of removing <i>keyword=value</i> (and <i>value@keyword</i> ) pairs and other general rules for list-oriented batch job attributes.                                                             |
| 32554 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 32555 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 32556 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 32557 | –V                           | Specify that all of the environment variables of the process are exported to the context of the batch job.                                                                                                                                                                                                                                                       |
| 32558 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 32559 |                              | The <i>qsub</i> utility shall place every environment variable in the process in which the utility is executing in the list and shall set the value of each variable in the attribute to the value of that variable in the process.                                                                                                                              |
| 32560 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 32561 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 32562 | –z                           | Specify that the utility does not write the batch <i>job_identifier</i> of the created batch job to standard output.                                                                                                                                                                                                                                             |
| 32563 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 32564 |                              | If the –z option is presented to the <i>qsub</i> utility, the utility shall not write the batch <i>job_identifier</i> of the created batch job to standard output.                                                                                                                                                                                               |
| 32565 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 32566 |                              | If the –z option is not presented to the <i>qsub</i> utility, the utility shall write the identifier of the created batch job to standard output.                                                                                                                                                                                                                |
| 32567 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 32568 | <b>OPERANDS</b>              |                                                                                                                                                                                                                                                                                                                                                                  |
| 32569 |                              | The <i>qsub</i> utility shall accept a <i>script</i> operand that indicates the path to the script of the batch job.                                                                                                                                                                                                                                             |
| 32570 |                              | If the <i>script</i> operand is not presented to the <i>qsub</i> utility, or if the operand is the single-character string ‘–’, the utility shall read the script from standard input.                                                                                                                                                                           |
| 32571 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 32572 |                              | If the script represents a partial path, the <i>qsub</i> utility shall expand the path relative to the current directory of the process executing the utility.                                                                                                                                                                                                   |
| 32573 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 32574 | <b>STDIN</b>                 |                                                                                                                                                                                                                                                                                                                                                                  |
| 32575 |                              | The <i>qsub</i> utility reads the script of the batch job from standard input if the script operand is omitted or is the single character ‘–’.                                                                                                                                                                                                                   |
| 32576 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 32577 | <b>INPUT FILES</b>           |                                                                                                                                                                                                                                                                                                                                                                  |
| 32578 |                              | In addition to binding the file indicated by the <i>script</i> operand to the batch job, the <i>qsub</i> utility reads the script file and acts on directives in the script.                                                                                                                                                                                     |
| 32579 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 32580 | <b>ENVIRONMENT VARIABLES</b> |                                                                                                                                                                                                                                                                                                                                                                  |
| 32581 |                              | The following environment variables shall affect the execution of <i>qsub</i> :                                                                                                                                                                                                                                                                                  |
| 32582 | <i>LANG</i>                  | Provide a default value for the internationalization variables that are unset or null. If <i>LANG</i> is unset or null, the corresponding value from the implementation-dependent default locale shall be used. If any of the internationalization variables contains an invalid setting, the utility shall behave as if none of the variables had been defined. |
| 32583 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 32584 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 32585 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 32586 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 32587 | <i>LC_ALL</i>                | If set to a non-empty string value, override the values of all the other internationalization variables.                                                                                                                                                                                                                                                         |
| 32588 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 32589 | <i>LC_CTYPE</i>              | Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).                                                                                                                                                                                        |
| 32590 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 32591 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 32592 | <i>LC_MESSAGES</i>           |                                                                                                                                                                                                                                                                                                                                                                  |
| 32593 |                              | Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.                                                                                                                                                                                                                                     |
| 32594 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 32595 | <i>LC_TIME</i>               | Determine the format and contents of date and time strings written by <i>qsub</i> .                                                                                                                                                                                                                                                                              |

|       |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------|-----------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 32596 | <b>LOGNAME</b>              | Determine the login name of the user.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 32597 | <b>PBS_DPREFIX</b>          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32598 |                             | Determine the default prefix for directives within the script.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 32599 | <b>SHELL</b>                | Determine the path name of the preferred command language interpreter of the user.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 32600 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32601 | <b>TZ</b>                   | Determine the timezone in which the time and date are written. If the <i>TZ</i> variable is not set, an unspecified system default timezone is used.                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 32602 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32603 | <b>ASYNCHRONOUS EVENTS</b>  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32604 |                             | Once created, a batch job exists until it exits, aborts, or is deleted.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 32605 |                             | After a batch job is created by the <i>qsub</i> utility, batch servers might route, execute, modify, or delete the batch job.                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 32606 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32607 | <b>STDOUT</b>               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32608 |                             | The <i>qsub</i> utility writes the batch <i>job_identifier</i> assigned to the batch job to standard output, unless the <i>-z</i> option is specified.                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 32609 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32610 | <b>STDERR</b>               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32611 |                             | Used only for diagnostic messages.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 32612 | <b>OUTPUT FILES</b>         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32613 |                             | None.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 32614 | <b>EXTENDED DESCRIPTION</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32615 | <b>Script Preservation</b>  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32616 |                             | The <i>qsub</i> utility shall make the script available to the server executing the batch job in such a way that the server executes the script as it exists at the time of submission.                                                                                                                                                                                                                                                                                                                                                                                                        |
| 32617 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32618 |                             | The <i>qsub</i> utility can send a copy of the script to the server with the <i>Queue Job Request</i> or store a temporary copy of the script in a location specified to the server.                                                                                                                                                                                                                                                                                                                                                                                                           |
| 32619 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32620 | <b>Option Specification</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32621 |                             | A script can contain directives to the <i>qsub</i> utility.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 32622 |                             | The <i>qsub</i> utility shall scan the lines of the script for directives, skipping blank lines, until the first line that begins with a string other than the directive string; if directives occur on subsequent lines, the utility shall ignore those directives.                                                                                                                                                                                                                                                                                                                           |
| 32623 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32624 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32625 |                             | Lines are separated by a <newline>. If the first line of the script begins with "#!" or a colon (' : '), then it is skipped. The <i>qsub</i> utility shall process a line in the script as a directive if, and only if, the string of characters from the first non-white-space character on the line until the first <space> or <tab> character on the line match the directive prefix. If a line in the script contains a directive and the final characters of the line are backslash ( ' \ ' ) and <newline>, then the next line shall be interpreted as a continuation of that directive. |
| 32626 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32627 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32628 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32629 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32630 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32631 |                             | The <i>qsub</i> utility shall process the options and option-arguments contained on the directive prefix line using the same syntax as if the options were input on the <i>qsub</i> utility.                                                                                                                                                                                                                                                                                                                                                                                                   |
| 32632 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32633 |                             | The <i>qsub</i> utility shall continue to process a directive prefix line until after a <newline> is encountered. An implementation may also ignore comments of the shell that will interpret the script. An implementation shall describe in the conformance document the format of any shell comments that it will recognize.                                                                                                                                                                                                                                                                |
| 32634 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32635 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 32636 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

32637 If an option is present in both a directive and the arguments to the *qsub* utility, the utility shall  
 32638 ignore the option and the corresponding option-argument, if any, in the directive.

32639 If an option that is present in the directive is not present in the arguments to the *qsub* utility, the  
 32640 utility shall process the option and the option-argument, if any.

32641 In order of preference, the *qsub* utility shall select the directive prefix from one of the following  
 32642 sources:

- 32643 • If the **-C** option is presented to the utility, the value of the *directive\_prefix* option-argument
- 32644 • If the environment variable *PBS\_DPREFIX* is defined, the value of that variable
- 32645 • The four-character string "#PBS" encoded in the portable character set

32646 If the **-C** option is present in the script file it shall be ignored.

#### 32647 **EXIT STATUS**

32648 The following exit values shall be returned:

- 32649 0 Successful completion.
- 32650 >0 An error occurred.

#### 32651 **CONSEQUENCES OF ERRORS**

32652 Default.

#### 32653 **APPLICATION USAGE**

32654 None.

#### 32655 **EXAMPLES**

32656 None.

#### 32657 **RATIONALE**

32658 The *qsub* utility allows users to create a batch job that will process the script specified as the  
 32659 operand of the utility.

32660 The options of the *qsub* utility allow users to control many aspects of the queuing and execution  
 32661 of a batch job.

32662 The **-a** option allows users to designate the time after which the batch job will become eligible to  
 32663 run. By specifying an execution time, users can take advantage of resources at off-peak hours,  
 32664 synchronize jobs with chronologically predictable events, and perhaps take advantage of off-  
 32665 peak pricing of computing time. For these reasons and others, a timing option is existing practice  
 32666 on the part of almost every batch system, including NQS.

32667 The **-A** option allows users to specify the account that will be charged for the batch job. Support  
 32668 for account is not mandatory for conforming batch servers.

32669 The **-C** option allows users to prescribe the prefix for directives within the script file. The default  
 32670 prefix "#PBS" may be inappropriate if the script will be interpreted with an alternate shell, as  
 32671 specified by the **-S** option.

32672 The **-c** option allows users to establish the checkpointing interval for their jobs. A checkpointing  
 32673 system, which is not defined by this volume of IEEE Std. 1003.1-200x, allows recovery of a batch  
 32674 job at the most recent checkpoint in the event of a crash. Checkpointing is typically used for jobs  
 32675 that consume expensive computing time or must meet a critical schedule. Users should be  
 32676 allowed to make the tradeoff between the overhead of checkpointing and the risk to the timely  
 32677 completion of the batch job; therefore, this volume of IEEE Std. 1003.1-200x provides the  
 32678 checkpointing interval option. Support for checkpointing is optional for batch servers.

32679 The **-e** option allows users to redirect the standard error streams of their jobs to a non-default  
32680 path. For example, if the submitted script generally produces a great deal of useless error output,  
32681 a user might redirect the standard error output to the null device. Or, if the file system holding  
32682 the default location (the home directory of the user) has too little free space, the user might  
32683 redirect the standard error stream to a file in another file system.

32684 The **-h** option allows users to create a batch job that is held until explicitly released. The ability  
32685 to create a held job is useful when some external event must complete before the batch job can  
32686 execute. For example, the user might submit a held job and release it when the system load has  
32687 dropped.

32688 The **-j** option allows users to merge the standard error of a batch job into its standard output  
32689 stream, which has the advantage of showing the sequential relationship between output and  
32690 error messages.

32691 The **-l** option allows users to limit the resources that will be consumed by the batch job. For  
32692 example, the user may wish to limit the amount of CPU time that can be consumed by a batch  
32693 job that has a risk of entering an infinite loop.

32694 The **-m** option allows users to designate those points in the execution of a batch job at which  
32695 mail will be sent to the submitting user, or to the account(s) indicated by the **-M** option. By  
32696 requesting mail notification at points of interest in the life of a job, the submitting user, or other  
32697 designated users, can track the progress of a batch job.

32698 The **-N** option allows users to associate a name with the batch job. The job name in no way  
32699 affects the processing of the batch job, but rather serves as a mnemonic handle for users. For  
32700 example, the batch job name can help the user distinguish between multiple jobs listed by the  
32701 *qstat* utility.

32702 The **-o** option allows users to redirect the standard output stream. A user might, for example,  
32703 wish to redirect to the null device the standard output stream of a job that produces copious yet  
32704 superfluous output.

32705 The **-P** option allows users to designate the relative priority of a batch job for selection from a  
32706 queue.

32707 The **-q** option allows users to specify an initial queue for the batch job. If the user specifies a  
32708 routing queue, the batch batch server routes the batch job to another queue for execution or  
32709 further routing. If the user specifies a non-routing queue, the batch server of the queue  
32710 eventually executes the batch job.

32711 The **-r** option allows users to control whether the submitted job will be rerun if the controlling  
32712 batch node fails during execution of the batch job. The **-r** option likewise allows users to  
32713 indicate whether or not the batch job is eligible to be rerun by the *qrerun* utility. Some jobs cannot  
32714 be correctly rerun because of changes they make in the state of databases or other aspects of  
32715 their environment. This volume of IEEE Std. 1003.1-200x specifies that the default, if the **-r**  
32716 option is not presented to the utility, will be that the batch job cannot be rerun, since the result of  
32717 rerunning a non-rerunable job might be catastrophic.

32718 The **-S** option allows users to specify the program (usually a shell) that will be invoked to  
32719 process the script of the batch job. This option has been modified to allow a list of shell names  
32720 and locations associated with different hosts.

32721 The **-u** option is useful when the submitting user is authorized to use more than one account on  
32722 a given host, in which case the **-u** option allows the user to select from among those accounts.  
32723 The option-argument is a list of user-host pairs, so that the submitting user can provide different  
32724 user identifiers for different nodes in the event the batch job is routed. The **-u** option provides a  
32725 lot of flexibility to accommodate sites with complex account structures. Users that have the

|       |                                                                                                             |
|-------|-------------------------------------------------------------------------------------------------------------|
| 32726 | same user identifier on all the hosts they are authorized to use will not need to use the <b>-u</b> option. |
| 32727 | The <b>-V</b> option allows users to export all their current environment variables, as of the time the     |
| 32728 | batch job is submitted, to the context of the processes of the batch job.                                   |
| 32729 | The <b>-v</b> option allows users to export specific environment variables from their current process       |
| 32730 | to the processes of the batch job.                                                                          |
| 32731 | The <b>-z</b> option allows users to suppress the writing of the batch job identifier to standard output.   |
| 32732 | The <b>-z</b> option is an existing NQS practice that has been standardized.                                |
| 32733 | As with other batch utilities, implementors can extend the <i>qsub</i> utility using the <b>-W</b> option.  |
| 32734 | Historically, the <i>qsub</i> utility has served the batch job-submission function in the NQS system, the   |
| 32735 | existing practice on which it is based. Some changes and additions have been made to the <i>qsub</i>        |
| 32736 | utility in this volume of IEEE Std. 1003.1-200x, <i>vis-a-vis</i> NQS, as a result of the growing pool of   |
| 32737 | experience with distributed batch systems.                                                                  |
| 32738 | The set of features of the <i>qsub</i> utility as defined in this volume of IEEE Std. 1003.1-200x appears   |
| 32739 | to incorporate all the common existing practice on potentially POSIX-conformant platforms.                  |
| 32740 | Where implementors wish to extend the functionality of their <i>qsub</i> utility, they may (as defined      |
| 32741 | by the base standard) use the <b>-W</b> option to provide implementation-dependent extensions.              |
| 32742 | <b>FUTURE DIRECTIONS</b>                                                                                    |
| 32743 | None.                                                                                                       |
| 32744 | <b>SEE ALSO</b>                                                                                             |
| 32745 | <i>qrerun</i> , <i>qstat</i> , <i>touch</i> , Chapter 3 on page 133                                         |
| 32746 | <b>CHANGE HISTORY</b>                                                                                       |
| 32747 | Derived from IEEE Std. 1003.2d-1994.                                                                        |



32748 **NAME**32749        *read* — read a line from standard input32750 **SYNOPSIS**32751        *read* [-r] *var*...32752 **DESCRIPTION**32753        The *read* utility shall read a single line from standard input.

32754        By default, unless the **-r** option is specified, backslash ('\**'**) shall act as an escape character, as  
 32755        described in Section 2.2.1 on page 36. If standard input is a terminal device and the invoking  
 32756        shell is interactive, *read* shall prompt for a continuation line when:

- 32757        • The shell reads an input line ending with a backslash, unless the **-r** option is specified.
- 32758        • A here-document is not terminated after a <newline> character is entered.

32759        The line shall be split into fields as in the shell (see Section 2.6.5 on page 58); the first field shall  
 32760        be assigned to the first variable *var*, the second field to the second variable *var*, and so on. If  
 32761        there are fewer *var* operands specified than there are fields, the leftover fields and their  
 32762        intervening separators shall be assigned to the last *var*. If there are fewer fields than *vars*, the  
 32763        remaining *vars* shall be set to empty strings.

32764        The setting of variables specified by the *var* operands shall affect the current shell execution  
 32765        environment; see Section 2.12 on page 90. If it is called in a subshell or separate utility execution  
 32766        environment, such as one of the following:

```
32767 (read foo)
32768 nohup read ...
32769 find . -exec read ... \;
```

32770        it shall not affect the shell variables in the caller's environment.

32771 **OPTIONS**

32772        The *read* utility shall conform to the System Interface Definitions volume of  
 32773        IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

32774        The following option is supported:

32775        **-r**        Do not treat a backslash character in any special way. Consider each backslash to  
 32776        be part of the input line.

32777 **OPERANDS**

32778        The following operand shall be supported:

32779        *var*        The name of an existing or nonexisting shell variable.

32780 **STDIN**

32781        The standard input shall be a text file.

32782 **INPUT FILES**

32783        None.

32784 **ENVIRONMENT VARIABLES**32785        The following environment variables shall affect the execution of *read*:

32786        *IFS*        Determine the internal field separators used to delimit fields; see Section 2.5.3 on  
 32787        page 45.

32788        *LANG*       Provide a default value for the internationalization variables that are unset or null.  
 32789        If *LANG* is unset or null, the corresponding value from the implementation-  
 32790        dependent default locale shall be used. If any of the internationalization variables

|       |                               |                                                                                                                        |
|-------|-------------------------------|------------------------------------------------------------------------------------------------------------------------|
| 32791 |                               | contains an invalid setting, the utility shall behave as if none of the variables had                                  |
| 32792 |                               | been defined.                                                                                                          |
| 32793 | <b>LC_ALL</b>                 | If set to a non-empty string value, override the values of all the other                                               |
| 32794 |                               | internationalization variables.                                                                                        |
| 32795 | <b>LC_CTYPE</b>               | Determine the locale for the interpretation of sequences of bytes of text data as                                      |
| 32796 |                               | characters (for example, single-byte as opposed to multi-byte characters in                                            |
| 32797 |                               | arguments).                                                                                                            |
| 32798 | <b>LC_MESSAGES</b>            |                                                                                                                        |
| 32799 |                               | Determine the locale that should be used to affect the format and contents of                                          |
| 32800 |                               | diagnostic messages written to standard error.                                                                         |
| 32801 | <b>XLSPATH</b>                | Determine the location of message catalogs for the processing of <b>LC_MESSAGES</b> .                                  |
| 32802 | <b>PS2</b>                    | Provide the prompt string that an interactive shell shall write to standard error                                      |
| 32803 |                               | when a line ending with a backslash is read and the <b>-r</b> option was not specified, or                             |
| 32804 |                               | if a here-document is not terminated after a <newline> character is entered.                                           |
| 32805 | <b>ASYNCHRONOUS EVENTS</b>    |                                                                                                                        |
| 32806 |                               | Default.                                                                                                               |
| 32807 | <b>STDOUT</b>                 |                                                                                                                        |
| 32808 |                               | Not used.                                                                                                              |
| 32809 | <b>STDERR</b>                 |                                                                                                                        |
| 32810 |                               | Used for diagnostic messages and prompts for continued input.                                                          |
| 32811 | <b>OUTPUT FILES</b>           |                                                                                                                        |
| 32812 |                               | None.                                                                                                                  |
| 32813 | <b>EXTENDED DESCRIPTION</b>   |                                                                                                                        |
| 32814 |                               | None.                                                                                                                  |
| 32815 | <b>EXIT STATUS</b>            |                                                                                                                        |
| 32816 |                               | The following exit values shall be returned:                                                                           |
| 32817 | 0                             | Successful completion.                                                                                                 |
| 32818 | >0                            | End-of-file was detected or an error occurred.                                                                         |
| 32819 | <b>CONSEQUENCES OF ERRORS</b> |                                                                                                                        |
| 32820 |                               | Default.                                                                                                               |
| 32821 | <b>APPLICATION USAGE</b>      |                                                                                                                        |
| 32822 |                               | The <i>read</i> utility has historically been a shell built-in.                                                        |
| 32823 |                               | The <b>-r</b> option is included to enable <i>read</i> to subsume the purpose of the <i>line</i> utility, which is not |
| 32824 |                               | included in IEEE Std. 1003.1-200x.                                                                                     |
| 32825 |                               | The results are undefined if an end-of-file is detected following a backslash at the end of a line                     |
| 32826 |                               | when <b>-r</b> is not specified.                                                                                       |
| 32827 | <b>EXAMPLES</b>               |                                                                                                                        |
| 32828 |                               | The following command:                                                                                                 |
| 32829 |                               | while read -r xx yy                                                                                                    |
| 32830 |                               | do                                                                                                                     |
| 32831 |                               | printf "%s %s\n" "\$yy" "\$xx"                                                                                         |
| 32832 |                               | done < <i>input_file</i>                                                                                               |

32833            prints a file with the first field of each line moved to the end of the line.

32834 **RATIONALE**

32835            The *read* utility historically has been a shell built-in. It was separated off into its own utility to  
32836            take advantage of the richer description of functionality introduced by this volume of  
32837            IEEE Std. 1003.1-200x.

32838            Since *read* affects the current shell execution environment, it is generally provided as a shell  
32839            regular built-in. If it is called in a subshell or separate utility execution environment, such as one  
32840            of the following:

```
32841 (read foo)
32842 nohup read ...
32843 find . -exec read ... \;
```

32844            it does not affect the shell variables in the environment of the caller.

32845 **FUTURE DIRECTIONS**

32846            None.

32847 **SEE ALSO**

32848            None.

32849 **CHANGE HISTORY**

32850            First released in Issue 2.

32851 **Issue 4**

32852            Relocated from the *sh* description for alignment with the ISO/IEC 9945-2: 1993 standard.

32853 **NAME**

32854           renice — set nice values of running processes

32855 **SYNOPSIS**32856 UP        renice -n *increment* [-g | -p | -u] *ID* ...

32857

32858 **DESCRIPTION**

32859       The *renice* utility shall request that the nice values (see the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 3.245, Nice Value) of one or more running processes be changed. By default, the applicable processes are specified by their process IDs. When a process group is specified (see -g), the request applies to all processes in the process group.

32863       The nice value shall be bounded in an implementation-dependent manner. If the requested *increment* would raise or lower the nice value of the executed utility beyond implementation-dependent limits, then the limit whose value was exceeded shall be used.

32866       When a user is *reniced*, the request applies to all processes whose saved set-user-ID matches the user ID corresponding to the user.

32868       Regardless of which options are supplied or any other factor, *renice* shall not alter the nice values of any process unless the user requesting such a change has appropriate privileges to do so for the specified process. If the user lacks appropriate privileges to perform the requested action, the utility shall return an error status.

32872       The saved set-user-ID of the user's process shall be checked instead of its effective user ID when *renice* attempts to determine the user ID of the process in order to determine whether the user has appropriate privileges.

32875 **OPTIONS**

32876       The *renice* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

32878       The following options shall be supported:

32879        -g           Interpret all operands as unsigned decimal integer process group IDs.

32880        -n *increment* Specify how the nice value of the specified process or processes is to be adjusted. The *increment* option-argument is a positive or negative decimal integer that shall be used to modify the nice value of the specified process or processes.

32883                   Positive *increment* values shall cause a lower nice value. Negative *increment* values may require appropriate privileges and shall cause a higher nice value.

32885        -p           Interpret all operands as unsigned decimal integer process IDs. The -p option is the default if no options are specified.

32887        -u           Interpret all operands as users. If a user exists with a user name equal to the operand, then the user ID of that user is used in further processing. Otherwise, if the operand represents an unsigned decimal integer, it shall be used as the numeric user ID of the user.

32891 **OPERANDS**

32892       The following operands shall be supported:

32893        *ID*           A process ID, process group ID, or user name/user ID, depending on the option selected.

32894

32895 **STDIN**

32896 Not used.

32897 **INPUT FILES**

32898 None.

32899 **ENVIRONMENT VARIABLES**32900 The following environment variables shall affect the execution of *renice*:

32901 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 32902 If *LANG* is unset or null, the corresponding value from the implementation-  
 32903 dependent default locale shall be used. If any of the internationalization variables  
 32904 contains an invalid setting, the utility shall behave as if none of the variables had  
 32905 been defined.

32906 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 32907 internationalization variables.

32908 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 32909 characters (for example, single-byte as opposed to multi-byte characters in  
 32910 arguments).

32911 *LC\_MESSAGES*

32912 Determine the locale that should be used to affect the format and contents of  
 32913 diagnostic messages written to standard error.

32914 *XSI NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

32915 **ASYNCHRONOUS EVENTS**

32916 Default.

32917 **STDOUT**

32918 Not used.

32919 **STDERR**

32920 Used only for diagnostic messages.

32921 **OUTPUT FILES**

32922 None.

32923 **EXTENDED DESCRIPTION**

32924 None.

32925 **EXIT STATUS**

32926 The following exit values shall be returned:

32927 0 Successful completion.

32928 &gt;0 An error occurred.

32929 **CONSEQUENCES OF ERRORS**

32930 Default.

32931 **APPLICATION USAGE**

32932 Application writers should note that this utility need not be provided on systems that do not  
 32933 support the User Portability Utilities option.

32934 **EXAMPLES**

32935 1. Adjust the nice value so that process IDs 987 and 32 would have a lower nice value:

32936 `renice -n 5 -p 987 32`

32937 2. Adjust the nice value so that group IDs 324 and 76 would have a higher nice value, if the  
 32938 user has the appropriate privileges to do so:

32939 `renice -n -4 -g 324 76`

32940 3. Adjust the nice value so that numeric user ID 8 and user **sas** would have a lower nice  
 32941 value:

32942 `renice -n 4 -u 8 sas`

32943 Useful nice value increments on historical systems include 19 or 20 (the affected processes run  
 32944 only when nothing else in the system attempts to run) and any negative number (to make  
 32945 processes run faster).

32946 **RATIONALE**

32947 The *gid*, *pid*, and *user* specifications do not fit either the definition of operand or option-  
 32948 argument. However, for clarity, they have been included in the OPTIONS section, rather than  
 32949 the OPERANDS section.

32950 The definition of nice value is not intended to suggest that all processes in a system have  
 32951 priorities that are comparable. Scheduling policy extensions such as the realtime priorities in  
 32952 POSIX.4 make the notion of a single underlying priority for all scheduling policies problematic.  
 32953 Some systems may implement the *nice* **-related** features to affect all processes on the system,  
 32954 others to affect just the general time-sharing activities implied by this volume of  
 32955 IEEE Std. 1003.1-200x, and others may have no effect at all. Because of the use of  
 32956 “implementation-dependent” in *nice* and *renice*, a wide range of implementation strategies are  
 32957 possible.

32958 Originally, this utility was written in the historical manner, using the term “nice value”. This  
 32959 was always a point of concern with users because it was never intuitively obvious what this  
 32960 meant. With a newer version of *renice*, which used the term “system scheduling priority”, it was  
 32961 hoped that novice users could better understand what this utility was meant to do. Also, it  
 32962 would be easier to document what the utility was meant to do. Unfortunately, the addition of  
 32963 the POSIX realtime scheduling capabilities introduced the concepts of process and thread  
 32964 scheduling priorities that were totally unaffected by the *nice*/*renice* utilities or the  
 32965 *nice()*/*setpriority()* functions. Continuing to use the term “system scheduling priority” would  
 32966 have incorrectly suggested that these utilities and functions were indeed affecting these realtime  
 32967 priorities. It was decided to revert to the historical term “nice value” to reference this unrelated  
 32968 process attribute.

32969 Although this utility has use by system administrators (and in fact appears in the system  
 32970 administration portion of the BSD documentation), the standard developers considered that it  
 32971 was very useful for individual end users to control their own processes.

32972 **FUTURE DIRECTIONS**

32973 None.

32974 **SEE ALSO**32975 *nice*32976 **CHANGE HISTORY**

32977 First released in Issue 4.

32978 **Issue 5**32979 In the SYNOPSIS, an ellipsis is added to the **-u** option in all three obsolescent forms.32980 **Issue 6**

32981 This utility is now marked as part of the User Portability Utilities option.

32982 The APPLICATION USAGE section is added.

32983 The obsolescent forms of the SYNOPSIS are removed.

32984 Text previously conditional on POSIX\_SAVED\_IDS is mandatory in this issue. This is a FIPS  
32985 requirement.

## 32986 NAME

32987 rm — remove directory entries

## 32988 SYNOPSIS

32989 rm [-fiRr] *file*...

## 32990 DESCRIPTION

32991 The *rm* utility shall remove the directory entry specified by each *file* argument.

32992 If either of the files dot or dot-dot are specified as the basename portion of an operand (that is,  
 32993 the final path name component), *rm* shall write a diagnostic message to standard error and do  
 32994 nothing more with such operands.

32995 For each *file* the following steps shall be taken:

- 32996 1. If the *file* does not exist:
    - 32997 a. If the **-f** option is not specified, write a diagnostic message to standard error.
    - 32998 b. Go on to any remaining *files*.
  - 32999 2. If *file* is of type directory, the following steps shall be taken:
    - 33000 a. If neither the **-R** option nor the **-r** option is specified, write a diagnostic message to  
 33001 standard error, do nothing more with *file*, and go on to any remaining files.
    - 33002 b. If the **-f** option is not specified, and either the permissions of *file* do not permit  
 33003 writing and the standard input is a terminal or the **-i** option is specified, write a  
 33004 prompt to standard error and read a line from the standard input. If the response is  
 33005 not affirmative, do nothing more with the current file and go on to any remaining  
 33006 files.
    - 33007 c. For each entry contained in *file*, other than dot or dot-dot, the four steps listed here  
 33008 (1-4) shall be taken with the entry as if it were a *file* operand. The *rm* utility shall not  
 33009 traverse directories by following symbolic links into other parts of the hierarchy, but  
 33010 shall remove the links themselves.
    - 33011 d. If the **-i** option is specified, write a prompt to standard error and read a line from the  
 33012 standard input. If the response is not affirmative, do nothing more with the current  
 33013 file, and go on to any remaining files.
  - 33014 3. If *file* is not of type directory, the **-f** option is not specified, and either the permissions of  
 33015 *file* do not permit writing and the standard input is a terminal or the **-i** option is specified,  
 33016 write a prompt to the standard error and read a line from the standard input. If the  
 33017 response is not affirmative, do nothing more with the current file and go on to any  
 33018 remaining files.
  - 33019 4. If the current file is a directory, *rm* shall perform actions equivalent to the *rmdir()* function  
 33020 defined in the System Interfaces volume of IEEE Std. 1003.1-200x called with a path name  
 33021 of the current file used as the *path* argument. If the current file is not a directory, *rm* shall  
 33022 perform actions equivalent to the *unlink()* function defined in the System Interfaces  
 33023 volume of IEEE Std. 1003.1-200x called with a path name of the current file used as the *path*  
 33024 argument.
- 33025 If this fails for any reason, *rm* shall write a diagnostic message to standard error, do  
 33026 nothing more with the current file, and go on to any remaining files.

33027 The *rm* utility shall be able to descend to arbitrary depths in a file hierarchy, and shall not fail  
 33028 due to path length limitations (unless an operand specified by the user exceeds system  
 33029 limitations).



33030 **OPTIONS**

33031 The *rm* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 33032 Section 12.2, Utility Syntax Guidelines.

33033 The following options shall be supported:

33034 **-f** Do not prompt for confirmation. Do not write diagnostic messages or modify the  
 33035 exit status in the case of nonexistent operands. Any previous occurrences of the **-i**  
 33036 option shall be ignored.

33037 **-i** Prompt for confirmation as described previously. Any previous occurrences of the  
 33038 **-f** option shall be ignored.

33039 **-R** Remove file hierarchies. See the DESCRIPTION.

33040 **-r** Equivalent to **-R**.

33041 **OPERANDS**

33042 The following operand shall be supported:

33043 *file* A path name of a directory entry to be removed.

33044 **STDIN**

33045 Used to read an input line in response to each prompt specified in the STDOUT section.  
 33046 Otherwise, the standard input shall not be used.

33047 **INPUT FILES**

33048 None.

33049 **ENVIRONMENT VARIABLES**

33050 The following environment variables shall affect the execution of *rm*:

33051 **LANG** Provide a default value for the internationalization variables that are unset or null.  
 33052 If **LANG** is unset or null, the corresponding value from the implementation-  
 33053 dependent default locale shall be used. If any of the internationalization variables  
 33054 contains an invalid setting, the utility shall behave as if none of the variables had  
 33055 been defined.

33056 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 33057 internationalization variables.

33058 **LC\_COLLATE**

33059 Determine the locale for the behavior of ranges, equivalence classes, and multi-  
 33060 character collating elements used in the extended regular expression defined for  
 33061 the **yesexpr** locale keyword in the **LC\_MESSAGES** category.

33062 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 33063 characters (for example, single-byte as opposed to multi-byte characters in  
 33064 arguments) and the behavior of character classes within regular expressions used  
 33065 in the extended regular expression defined for the **yesexpr** locale keyword in the  
 33066 **LC\_MESSAGES** category.

33067 **LC\_MESSAGES**

33068 Determine the locale for the processing of affirmative responses that should be  
 33069 used to affect the format and contents of diagnostic messages written to standard  
 33070 error.

33071 **XS1** **NLSPATH** Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

33072 **ASYNCHRONOUS EVENTS**

33073 Default.

33074 **STDOUT**

33075 Not used.

33076 **STDERR**

33077 Prompts shall be written to standard error under the conditions specified in the DESCRIPTION  
 33078 and OPTIONS sections. The prompts shall contain the *file* path name, but their format is  
 33079 otherwise unspecified. The standard error also shall be used for diagnostic messages.

33080 **OUTPUT FILES**

33081 None.

33082 **EXTENDED DESCRIPTION**

33083 None.

33084 **EXIT STATUS**

33085 The following exit values shall be returned:

33086 0 All of the named directory entries for which *rm* performed actions equivalent to *rmdir()* or  
 33087 *unlink()* functions were removed.

33088 &gt;0 An error occurred.

33089 **CONSEQUENCES OF ERRORS**

33090 Default.

33091 **APPLICATION USAGE**

33092 The *rm* utility is forbidden to remove the names dot and dot-dot in order to avoid the  
 33093 consequences of inadvertently doing something like:

33094 `rm -r .*`

33095 Some systems do not permit the removal of the last link to an executable binary file that is being  
 33096 executed; see the [EBUSY] error in the *unlink()* function defined in the System Interfaces volume  
 33097 of IEEE Std. 1003.1-200x. Thus, the *rm* utility can fail to remove such files.

33098 The *-i* option causes *rm* to prompt and read the standard input even if the standard input is not  
 33099 a terminal, but in the absence of *-i* the mode prompting is not done when the standard input is  
 33100 not a terminal.

33101 **EXAMPLES**

33102 1. The following command:

33103 `rm a.out core`33104 removes the directory entries: **a.out** and **core**.

33105 2. The following command:

33106 `rm -Rf junk`33107 removes the directory **junk** and all its contents, without prompting.33108 **RATIONALE**

33109 The *-i* option causes *rm* to prompt and read the standard input even if the standard input is not  
 33110 a terminal, but, in the absence of *-i*, the mode prompting is not done when the standard input is  
 33111 not a terminal.

33112 For absolute clarity, paragraphs (2b) and (3) in the DESCRIPTION of *rm* describing the behavior  
 33113 when prompting for confirmation, should be interpreted in the following manner:

```

33114 if ((NOT f_option) AND
33115 ((not_writable AND input_is_terminal) OR i_option))

```

33116 The exact format of the interactive prompts is unspecified. Only the general nature of the  
 33117 contents of prompts are specified because implementations may desire more descriptive  
 33118 prompts than those used on historical implementations. Therefore, an application not using the  
 33119 `-f` option, or using the `-i` option, relies on the system to provide the most suitable dialog directly  
 33120 with the user, based on the behavior specified.

33121 The `-r` option is historical practice on all known systems. The synonym `-R` option is provided  
 33122 for consistency with the other utilities in this volume of IEEE Std. 1003.1-200x that provide  
 33123 options requesting recursive descent through the file hierarchy.

33124 The behavior of the `-f` option in historical versions of *rm* is inconsistent. In general, along with  
 33125 “forcing” the unlink without prompting for permission, it always causes diagnostic messages to  
 33126 be suppressed and the exit status to be unmodified for nonexistent operands and files that  
 33127 cannot be unlinked. In some versions, however, the `-f` option suppresses usage messages and  
 33128 system errors as well. Suppressing such messages is not a service to either shell scripts or users.

33129 It is less clear that error messages regarding unlinkable files should be suppressed. Although this  
 33130 is historical practice, this volume of IEEE Std. 1003.1-200x does not permit the `-f` option to  
 33131 suppress such messages.

33132 When given the `-r` and `-i` options, historical versions of *rm* prompt the user twice for each  
 33133 directory, once before removing its contents and once before actually attempting to delete the  
 33134 directory entry that names it. This allows the user to “prune” the file hierarchy walk. Historical  
 33135 versions of *rm* were inconsistent in that some did not do the former prompt for directories  
 33136 named on the command line and others had obscure prompting behavior when the `-i` option  
 33137 was specified and the permissions of the file did not permit writing. The POSIX Shell and  
 33138 Utilities *rm* differs little from historic practice, but does require that prompts be consistent.  
 33139 Historical versions of *rm* were also inconsistent in that prompts were done to both standard  
 33140 output and standard error. This volume of IEEE Std. 1003.1-200x requires that prompts be done  
 33141 to standard error, for consistency with *cp* and *mv*, and to allow historical extensions to *rm* that  
 33142 provide an option to list deleted files on standard output.

33143 The *rm* utility is required to descend to arbitrary depths so that any file hierarchy may be  
 33144 deleted. This means, for example, that the *rm* utility cannot run out of file descriptors during its  
 33145 descent (that is, if the number of file descriptors is limited, *rm* cannot be implemented in the  
 33146 historical fashion where one file descriptor is used per directory level). Also, *rm* is not permitted  
 33147 to fail because of path length restrictions, unless an operand specified by the user is longer than  
 33148 {PATH\_MAX}.

33149 The *rm* utility removes symbolic links themselves, not the files they refer to, as a consequence of  
 33150 the dependence on the *unlink()* functionality, per the DESCRIPTION. When removing  
 33151 hierarchies with `-r` or `-R`, the prohibition on following symbolic links has to be made explicit.

#### 33152 FUTURE DIRECTIONS

33153 None.

#### 33154 SEE ALSO

33155 *rmdir*, the System Interfaces volume of IEEE Std. 1003.1-200x, *remove()*, *unlink()*

#### 33156 CHANGE HISTORY

33157 First released in Issue 2.

|       |                |                                                                                                     |
|-------|----------------|-----------------------------------------------------------------------------------------------------|
| 33158 | <b>Issue 4</b> |                                                                                                     |
| 33159 |                | Aligned with the ISO/IEC 9945-2: 1993 standard.                                                     |
| 33160 | <b>Issue 5</b> |                                                                                                     |
| 33161 |                | FUTURE DIRECTIONS section added.                                                                    |
| 33162 | <b>Issue 6</b> |                                                                                                     |
| 33163 |                | Text is added to clarify actions relating to symbolic links as specified in the IEEE P1003.2b draft |
| 33164 |                | standard.                                                                                           |

33165 **NAME**33166       rm del — remove a delta from an SCCS file (**DEVELOPMENT**)33167 **SYNOPSIS**33168 xSI       rm del -r *SID file...*

33169

33170 **DESCRIPTION**

33171       The *rm del* utility shall remove the delta specified by the *SID* from each named SCCS file. The  
 33172       delta to be removed shall be the most recent delta in its branch in the delta chain of each named  
 33173       SCCS file. In addition, the application shall ensure that the *SID* specified is not that of a version  
 33174       being edited for the purpose of making a delta; that is, if a *p-file* (see *get* on page 510) exists for  
 33175       the named SCCS file, the *SID* specified shall not appear in any entry of the *p-file*.

33176       Removal of a delta is restricted to:

- 33177           1. The user who made the delta
- 33178           2. The owner of the SCCS file
- 33179           3. The owner of the directory containing the SCCS file

33180 **OPTIONS**

33181       The *rm del* utility shall conform to the System Interface Definitions volume of  
 33182       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

33183       The following option shall be supported:

33184       -r *SID*       Specify the SCCS identification string (*SID*) of the delta to be deleted.33185 **OPERANDS**

33186       The following operand shall be supported:

33187       *file*       A path name of an existing SCCS file or a directory. If *file* is a directory, *rm del*  
 33188       behaves as though each file in the directory were specified as a named file, except  
 33189       that non-SCCS files (last component of the path name does not begin with **s.**) and  
 33190       unreadable files are silently ignored.

33191       If a single instance *file* is specified as *'-'*, the standard input is read; each line of  
 33192       the standard input is taken to be the name of an SCCS file to be processed. Non-  
 33193       SCCS files and unreadable files are silently ignored.

33194 **STDIN**

33195       The standard input shall be a text file used only when the *file* operand is specified as *'-'*. Each  
 33196       line of the text file shall be interpreted as an SCCS path name.

33197 **INPUT FILES**

33198       The SCCS files are files of unspecified format.

33199 **ENVIRONMENT VARIABLES**33200       The following environment variables shall affect the execution of *rm del*:

33201       *LANG*       Provide a default value for the internationalization variables that are unset or null.  
 33202       If *LANG* is unset or null, the corresponding value from the implementation-  
 33203       dependent default locale shall be used. If any of the internationalization variables  
 33204       contains an invalid setting, the utility shall behave as if none of the variables had  
 33205       been defined.

33206       *LC\_ALL*       If set to a non-empty string value, override the values of all the other  
 33207       internationalization variables.

- 33208 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 33209 characters (for example, single-byte as opposed to multi-byte characters in  
 33210 arguments and input files).
- 33211 **LC\_MESSAGES**  
 33212 Determine the locale that should be used to affect the format and contents of  
 33213 diagnostic messages written to standard error.
- 33214 **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.
- 33215 **ASYNCHRONOUS EVENTS**  
 33216 Default.
- 33217 **STDOUT**  
 33218 Not used.
- 33219 **STDERR**  
 33220 Used only for diagnostic messages.
- 33221 **OUTPUT FILES**  
 33222 The SCCS files are files of unspecified format. During processing of a *file*, a temporary *x-file*, as  
 33223 described in *admin* on page 160, may be created and deleted; a locking *z-file*, as described in *get*  
 33224 on page 510, may be created and deleted.
- 33225 **EXTENDED DESCRIPTION**  
 33226 None.
- 33227 **EXIT STATUS**  
 33228 The following exit values shall be returned:  
 33229 0 Successful completion.  
 33230 >0 An error occurred.
- 33231 **CONSEQUENCES OF ERRORS**  
 33232 Default.
- 33233 **APPLICATION USAGE**  
 33234 None.
- 33235 **EXAMPLES**  
 33236 None.
- 33237 **RATIONALE**  
 33238 None.
- 33239 **FUTURE DIRECTIONS**  
 33240 None.
- 33241 **SEE ALSO**  
 33242 *delta, get, prs*
- 33243 **CHANGE HISTORY**  
 33244 First released in Issue 2.
- 33245 **Issue 4**  
 33246 Format reorganized.  
 33247 Utility Syntax Guidelines support mandated.  
 33248 Internationalized environment variable support mandated.

33249 **Issue 6**

33250 The normative text is reworded to avoid use of the term “must” for application requirements.

33251 **NAME**

33252           rmdir — remove directories

33253 **SYNOPSIS**33254           rmdir [-p] *dir...*33255 **DESCRIPTION**

33256           The *rmdir* utility shall remove the directory entry specified by each *dir* operand, which the  
 33257           application shall ensure refers to an empty directory.

33258           Directories shall be processed in the order specified. If a directory and a subdirectory of that  
 33259           directory are specified in a single invocation of the *rmdir* utility, the application shall specify the  
 33260           subdirectory before the parent directory so that the parent directory will be empty when the  
 33261           *rmdir* utility tries to remove it.

33262 **OPTIONS**

33263           The *rmdir* utility shall conform to the System Interface Definitions volume of  
 33264           IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

33265           The following option shall be supported:

33266           **-p**           Remove all directories in a path name. For each *dir* operand:

- 33267                       1. The directory entry it names shall be removed.
- 33268                       2. If the *dir* operand includes more than one path name component, effects  
 33269                       equivalent to the following command shall occur:

33270                       rmdir -p \$(dirname *dir*)

33271 **OPERANDS**

33272           The following operand shall be supported:

33273           *dir*           A path name of an empty directory to be removed.

33274 **STDIN**

33275           Not used.

33276 **INPUT FILES**

33277           None.

33278 **ENVIRONMENT VARIABLES**

33279           The following environment variables shall affect the execution of *rmdir*:

33280           **LANG**       Provide a default value for the internationalization variables that are unset or null.  
 33281                       If *LANG* is unset or null, the corresponding value from the implementation-  
 33282                       dependent default locale shall be used. If any of the internationalization variables  
 33283                       contains an invalid setting, the utility shall behave as if none of the variables had  
 33284                       been defined.

33285           **LC\_ALL**      If set to a non-empty string value, override the values of all the other  
 33286                       internationalization variables.

33287           **LC\_CTYPE**   Determine the locale for the interpretation of sequences of bytes of text data as  
 33288                       characters (for example, single-byte as opposed to multi-byte characters in  
 33289                       arguments).

33290           **LC\_MESSAGES**

33291                       Determine the locale that should be used to affect the format and contents of  
 33292                       diagnostic messages written to standard error.



33293 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

### 33294 ASYNCHRONOUS EVENTS

33295 Default.

### 33296 STDOUT

33297 Not used.

### 33298 STDERR

33299 Used only for diagnostic messages.

### 33300 OUTPUT FILES

33301 None.

### 33302 EXTENDED DESCRIPTION

33303 None.

### 33304 EXIT STATUS

33305 The following exit values shall be returned:

33306 0 Each directory entry specified by a *dir* operand was removed successfully.

33307 >0 An error occurred.

### 33308 CONSEQUENCES OF ERRORS

33309 Default.

### 33310 APPLICATION USAGE

33311 The definition of an empty directory is one that contains, at most, directory entries for dot and  
33312 dot-dot.

### 33313 EXAMPLES

33314 If a directory **a** in the current directory is empty except it contains a directory **b** and **a/b** is empty  
33315 except it contains a directory **c**:

33316 `rmdir -p a/b/c`

33317 removes all three directories.

### 33318 RATIONALE

33319 On historical System V systems, the **-p** option also caused a message to be written to the  
33320 standard output. The message indicated whether the whole path was removed or whether part  
33321 of the path remained for some reason. The STDERR section requires this diagnostic when the  
33322 entire path specified by a *dir* operand is not removed, but does not allow the status message  
33323 reporting success to be written as a diagnostic.

33324 The *rmdir* utility on System V also included an **-s** option that suppressed the informational  
33325 message output by the **-p** option. This option has been omitted because the informational  
33326 message is not specified by this volume of IEEE Std. 1003.1-200x.

### 33327 FUTURE DIRECTIONS

33328 None.

### 33329 SEE ALSO

33330 *rm*, the System Interfaces volume of IEEE Std. 1003.1-200x, *remove()*, *rmdir()*, *unlink()*

### 33331 CHANGE HISTORY

33332 First released in Issue 2.

|       |                                                                                              |  |
|-------|----------------------------------------------------------------------------------------------|--|
| 33333 | <b>Issue 4</b>                                                                               |  |
| 33334 | Separated from the <i>rm</i> description and aligned with the ISO/IEC 9945-2: 1993 standard. |  |
| 33335 | <b>Issue 6</b>                                                                               |  |
| 33336 | The normative text is reworded to avoid use of the term “must” for application requirements. |  |

33337 **NAME**33338           sact — print current SCCS file-editing activity (**DEVELOPMENT**)33339 **SYNOPSIS**33340 XSI       sact *file...*

33341

33342 **DESCRIPTION**

33343       The *sact* utility shall inform the user of any impending deltas to a named SCCS file by writing a  
 33344       list to standard output. This situation occurs when *get -e* has been executed previously without  
 33345       a subsequent execution of *delta*.

33346 **OPTIONS**

33347       None.

33348 **OPERANDS**

33349       The following operand shall be supported:

33350       *file*       A path name of an existing SCCS file or a directory. If *file* is a directory, *sact*  
 33351       behaves as though each file in the directory were specified as a named file, except  
 33352       that non-SCCS files (last component of the path name does not begin with *s.*) and  
 33353       unreadable files are silently ignored.

33354       If a single instance *file* is specified as *'-'*, the standard input is read; each line of  
 33355       the standard input shall be taken to be the name of an SCCS file to be processed.  
 33356       Non-SCCS files and unreadable files shall be silently ignored.

33357 **STDIN**

33358       The standard input shall be a text file used only when the *file* operand is specified as *'-'*. Each  
 33359       line of the text file shall be interpreted as an SCCS path name.

33360 **INPUT FILES**

33361       Any SCCS files interrogated are files of an unspecified format.

33362 **ENVIRONMENT VARIABLES**33363       The following environment variables shall affect the execution of *sact*:

33364       *LANG*       Provide a default value for the internationalization variables that are unset or null.  
 33365       If *LANG* is unset or null, the corresponding value from the implementation-  
 33366       dependent default locale shall be used. If any of the internationalization variables  
 33367       contains an invalid setting, the utility shall behave as if none of the variables had  
 33368       been defined.

33369       *LC\_ALL*      If set to a non-empty string value, override the values of all the other  
 33370       internationalization variables.

33371       *LC\_CTYPE*   Determine the locale for the interpretation of sequences of bytes of text data as  
 33372       characters (for example, single-byte as opposed to multi-byte characters in  
 33373       arguments and input files).

33374       *LC\_MESSAGES*

33375       Determine the locale that should be used to affect the format and contents of  
 33376       diagnostic messages written to standard error.

33377       *NLSPATH*     Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

33378 **ASYNCHRONOUS EVENTS**

33379 Default.

33380 **STDOUT**

33381 The output for each named file shall consist of a line in the following format:

33382 "%sΔ%sΔ%sΔ%sΔ%s\n", &lt;SID&gt;, &lt;new SID&gt;, &lt;login&gt;, &lt;date&gt;, &lt;time&gt;

33383 <SID> Specifies the SID of a delta that currently exists in the SCCS file to which changes  
 33384 are made to make the new delta.

33385 <new SID> Specifies the SID for the new delta to be created.

33386 <login> Contains the login name of the user who makes the delta (that is, who executed a  
 33387 *get* for editing).

33388 <date> Contains the date that *get -e* was executed, in the format used by the *prs :D:* data  
 33389 keyword.

33390 <time> Contains the time that *get -e* was executed, in the format used by the *prs :T:* data  
 33391 keyword.

33392 If there is more than one named file or if a directory or standard input is named, each path name  
 33393 shall be written before each of the preceding lines:

33394 "\n%s:\n", &lt;pathname&gt;

33395 **STDERR**

33396 Used only for optional informative messages concerning SCCS files with no impending deltas,  
 33397 and for diagnostic messages.

33398 **OUTPUT FILES**

33399 None.

33400 **EXTENDED DESCRIPTION**

33401 None.

33402 **EXIT STATUS**

33403 The following exit values shall be returned:

33404 0 Successful completion.

33405 &gt;0 An error occurred.

33406 **CONSEQUENCES OF ERRORS**

33407 Default.

33408 **APPLICATION USAGE**

33409 None.

33410 **EXAMPLES**

33411 None.

33412 **RATIONALE**

33413 None.

33414 **FUTURE DIRECTIONS**

33415 None.

33416 **SEE ALSO**33417 *delta, get, unget*33418 **CHANGE HISTORY**

33419 First released in Issue 2.

33420 **Issue 4**

33421 Format reorganized.

33422 Utility Syntax Guidelines support mandated.

33423 Internationalized environment variable support mandated.

33424 **Issue 4, Version 2**33425 The `STDERR` section encompasses informative messages concerning SCCS files with no  
33426 impending deltas.

## 33427 NAME

33428        *sccs* — front end for the SCCS subsystem (**DEVELOPMENT**)

## 33429 SYNOPSIS

33430 XSI        *sccs* [*-r*][*-d path*][*-p path*] *command* [*options...*][*operands...*]

33431

## 33432 DESCRIPTION

33433        The *sccs* utility is a front end to the SCCS programs. It also includes the capability to run set-  
33434 user-id to another user to provide additional protection.33435        The *sccs* utility shall invoke the specified *command* with the specified *options* and *operands*. By  
33436 default, each of the *operands* shall be modified by prefixing it with the string **SCCS/s..**.33437        The *command* operand can be one of the SCCS utilities in this volume of IEEE Std. 1003.1-200x  
33438 (*admin*, *delta*, *get*, *prs*, *rmdel*, *sact*, *unget*, *val*, or *what*) or one of the pseudo-utilities listed in the  
33439 EXTENDED DESCRIPTION section.

## 33440 OPTIONS

33441        The *sccs* utility shall conform to the System Interface Definitions volume of  
33442 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines, except that *options* operands are  
33443 actually options to be passed to the utility named by *command*. When the portion of the  
33444 command:33445        *command* [*options ...*] [*operands ...*]33446        is considered, all of the pseudo-utilities used as *command* shall support the Utility Syntax  
33447 Guidelines. Any of the other SCCS utilities that can be invoked in this manner support the  
33448 Guidelines to the extent indicated by their individual OPTIONS sections.33449        The following options shall be supported preceding the *command* operand:33450        **-d path**        A path name of a directory to be used as a root directory for the SCCS files. The  
33451                        default is the current directory. The **-d** option takes precedence over the  
33452                        *PROJECTDIR* variable. See **-p**.33453        **-p path**        A path name of a directory in which the SCCS files are located. The default is the  
33454                        **SCCS** directory.33455        The **-p** option differs from the **-d** option in that the **-d** option-argument is  
33456                        prefixed to the entire path name and the **-p** option-argument is inserted before the  
33457                        final component of the path name. For example:33458        *sccs -d /x -p y get a/b*

33459        converts to:

33460        *get /x/a/y/s.b*

33461        This allows the creation of aliases such as:

33462        *alias syssccs="sccs -d /usr/src"*

33463        which is used as:

33464        *syssccs get cmd/who.c*33465        **-r**                Invoke *command* with the real user ID of the process, not any effective user ID that  
33466                        the *sccs* utility is set to. Certain commands (*admin*, **check**, **clean**, **diffs**, **info**, *rmdel*,  
33467                        and **tell**) cannot be run set-user-ID by all users, since this would allow anyone to  
33468                        change the authroizations. These commands are always run as the real user.

33469 **OPERANDS**

33470 The following operands shall be supported:

33471 *command* An SCCS utility name or the name of one of the pseudo-utilities listed in the  
 33472 EXTENDED DESCRIPTION section.

33473 *options* An option or option-argument to be passed to *command*.

33474 *operands* An operand to be passed to *command*.

33475 **STDIN**33476 See the utility description for the specified *command*.33477 **INPUT FILES**33478 See the utility description for the specified *command*.33479 **ENVIRONMENT VARIABLES**33480 The following environment variables shall affect the execution of *sccs*:

33481 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 33482 If *LANG* is unset or null, the corresponding value from the implementation-  
 33483 dependent default locale shall be used. If any of the internationalization variables  
 33484 contains an invalid setting, the utility shall behave as if none of the variables had  
 33485 been defined.

33486 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 33487 internationalization variables.

33488 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 33489 characters (for example, single-byte as opposed to multi-byte characters in  
 33490 arguments and input files).

33491 *LC\_MESSAGES*

33492 Determine the locale that should be used to affect the format and contents of  
 33493 diagnostic messages written to standard error.

33494 *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

33495 *PROJECTDIR*

33496 Provide a default value for the *-d path* option. If the value of *PROJECTDIR* begins  
 33497 with a slash, it shall be considered an absolute path name; otherwise, the value of  
 33498 *PROJECTDIR* is treated as a user name and that user's initial working directory  
 33499 shall be examined for a subdirectory **src** or **source**. If such a directory is found, it  
 33500 shall be used. Otherwise, the value shall be used as a relative path name.

33501 Additional environment variable effects may be found in the utility description for the specified  
 33502 *command*.

33503 **ASYNCHRONOUS EVENTS**

33504 Default.

33505 **STDOUT**33506 See the utility description for the specified *command*.33507 **STDERR**33508 See the utility description for the specified *command*.

33509 **OUTPUT FILES**

33510 See the utility description for the specified *command*.

33511 **EXTENDED DESCRIPTION**

33512 The following pseudo-utilities are supported as *command* operands. All options referred to in the  
 33513 following list are values given in the *options* operands following *command*.

33514 **check** Equivalent to **info**, except that nothing is printed if nothing is being edited, and a non-  
 33515 zero exit status is returned if anything is being edited. The intent is to have this  
 33516 included in an “install” entry in a makefile to ensure that everything is included into  
 33517 the SCCS file before a version is installed.

33518 **clean** Remove everything from the current directory that can be recreated from SCCS files,  
 33519 but do not remove any files being edited. If the **-b** option is given, branches are ignored  
 33520 in the determination of whether they are being edited; this is dangerous if branches are  
 33521 kept in the same directory.

33522 **create** Create an SCCS file, taking the initial contents from the file of the same name. Any  
 33523 options to *admin* are accepted. If the creation is successful, the original files are  
 33524 renamed by prefixing the basenames with a comma. These renamed files should be  
 33525 removed after it has been verified that the SCCS files have been created successfully.

33526 **delget** Perform a *delta* on the named files and then *get* new versions. The new versions have ID  
 33527 keywords expanded and are not editable. Any **-m**, **-p**, **-r**, **-s**, and **-y** options are passed  
 33528 to *delta*, and any **-b**, **-c**, **-e**, **-i**, **-k**, **-l**, **-s**, and **-x** options are passed to *get*.

33529 **deledit** Equivalent to **delget**, except that the *get* phase includes the **-e** option. This option is  
 33530 useful for making a checkpoint of the current editing phase. The same options are  
 33531 passed to *delta* as described above, and all the options listed for *get* above except **-e** are  
 33532 passed to *edit*.

33533 **diffs** Write a difference listing between the current version of the files checked out for  
 33534 editing and the versions in SCCS format. Any **-r**, **-c**, **-i**, **-x**, and **-t** options are passed to  
 33535 *get*; any **-l**, **-s**, **-e**, **-f**, **-h**, and **-b** options are passed to *diff*. A **-C** option is passed to *diff*  
 33536 as **-c**.

33537 **edit** Equivalent to *get -e*.

33538 **fix** Remove the named delta, but leave a copy of the delta with the changes that were in it.  
 33539 It is useful for fixing small compiler bugs, and so on. The application shall ensure that it  
 33540 is followed by a **-r** *SID* option. Since **fix** doesn't leave audit trails, it should be used  
 33541 carefully.

33542 **info** Write a listing of all files being edited. If the **-b** option is given, branches (that is, SIDs  
 33543 with two or fewer components) are ignored. If a **-u** *user* option is given, then only files  
 33544 being edited by the named user are listed. A **-U** option is equivalent to  
 33545 **-u**<current user>.

33546 **print** Write out verbose information about the named files, equivalent to *sccs prs*.

33547 **tell** Write a <newline>-separated list of the files being edited to standard output. Takes the  
 33548 **-b**, **-u**, and **-U** options like **info** and **check**.

33549 **unedit** This is the opposite of an **edit** or a *get -e*. It should be used with caution, since any  
 33550 changes made since the *get* are lost.



33551 **EXIT STATUS**

33552 The following exit values shall be returned:

33553 0 Successful completion.

33554 &gt;0 An error occurred.

33555 **CONSEQUENCES OF ERRORS**

33556 Default.

33557 **APPLICATION USAGE**

33558 Many of the SCCS utilities take directory names as operands as well as specific file names. The  
 33559 pseudo-utilities supported by *sccs* are not described as having this capability, but are not  
 33560 prohibited from doing so.

33561 **EXAMPLES**

33562 1. To get a file for editing, edit it and produce a new delta:

33563 `sccs get -e file.c`33564 `ex file.c`33565 `sccs delta file.c`

33566 2. To get a file from another directory:

33567 `sccs -p /usr/src/sccs/s. get cc.c`

33568 or:

33569 `sccs get /usr/src/sccs/s.cc.c`

33570 3. To make a delta of a large number of files in the current directory:

33571 `sccs delta *.c`

33572 4. To get a list of files being edited that are not on branches:

33573 `sccs info -b`

33574 5. To delta everything being edited by the current user:

33575 `sccs delta $(sccs tell -U)`

33576 6. In a makefile, to get source files from an SCCS file if it does not already exist:

33577 `SRCS = <list of source files>`33578 `$(SRCS):`33579 `sccs get $(REL) $@`33580 **RATIONALE**

33581 None.

33582 **FUTURE DIRECTIONS**

33583 None.

33584 **SEE ALSO**33585 *admin, delta, get, make, prs, rmdel, sact, unget, val, what*33586 **CHANGE HISTORY**

33587 First released in Issue 4.

33588 **Issue 6**

33589 In the ENVIRONMENT VARIABLES section, the *PROJECTDIR* description is updated from  
33590 “otherwise, the home directory of a user of that name is examined” to “otherwise, the value of  
33591 *PROJECTDIR* is treated as a user name and that user’s initial working directory is examined”.

33592 The normative text is reworded to avoid use of the term “must” for application requirements.

33593 **NAME**33594 `sed` — stream editor33595 **SYNOPSIS**33596 `sed [-n] script[file...]`33597 `sed [-n][-e script...[-f script_file]...[file...]`33598 **DESCRIPTION**

33599 The *sed* utility is a stream editor that shall read one or more text files, make editing changes  
 33600 according to a script of editing commands, and write the results to standard output. The script  
 33601 shall be obtained from either the *script* operand string or a combination of the option-arguments  
 33602 from the `-e script` and `-f script_file` options.

33603 **OPTIONS**

33604 The *sed* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 33605 Section 12.2, Utility Syntax Guidelines, except that the order of presentation of the `-e` and `-f`  
 33606 options is significant.

33607 The following options shall be supported:

33608 `-e script` Add the editing commands specified by the *script* option-argument to the end of  
 33609 the script of editing commands. The *script* option-argument shall have the same  
 33610 properties as the *script* operand, described in the OPERANDS section.

33611 `-f script_file` Add the editing commands in the file *script\_file* to the end of the script.

33612 `-n` Suppress the default output (in which each line, after it is examined for editing, is  
 33613 written to standard output). Only lines explicitly selected for output are written.

33614 Multiple `-e` and `-f` options may be specified. All commands shall be added to the script in the  
 33615 order specified, regardless of their origin.

33616 **OPERANDS**

33617 The following operands shall be supported:

33618 *file* A path name of a file whose contents are read and edited. If multiple *file* operands  
 33619 are specified, the named files shall be read in the order specified and the  
 33620 concatenation shall be edited. If no *file* operands are specified, the standard input  
 33621 shall be used.

33622 *script* A string to be used as the script of editing commands. The application shall not  
 33623 present a *script* that violates the restrictions of a text file except that the final  
 33624 character need not be a <newline> character.

33625 **STDIN**

33626 The standard input shall be used only if no *file* operands are specified. See the INPUT FILES  
 33627 section.

33628 **INPUT FILES**

33629 The input files shall be text files. The *script\_files* named by the `-f` option shall consist of editing  
 33630 commands.

33631 **ENVIRONMENT VARIABLES**

33632 The following environment variables shall affect the execution of *sed*:

33633 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 33634 If *LANG* is unset or null, the corresponding value from the implementation-  
 33635 dependent default locale shall be used. If any of the internationalization variables  
 33636 contains an invalid setting, the utility shall behave as if none of the variables had

33637                   been defined.

33638       **LC\_ALL**       If set to a non-empty string value, override the values of all the other  
33639                   internationalization variables.

33640       **LC\_COLLATE**  
33641                   Determine the locale for the behavior of ranges, equivalence classes, and multi-  
33642                   character collating elements within regular expressions.

33643       **LC\_CTYPE**   Determine the locale for the interpretation of sequences of bytes of text data as  
33644                   characters (for example, single-byte as opposed to multi-byte characters in  
33645                   arguments and input files), and the behavior of character classes within regular  
33646                   expressions.

33647       **LC\_MESSAGES**  
33648                   Determine the locale that should be used to affect the format and contents of  
33649                   diagnostic messages written to standard error.

33650 XSI       **NLSPATH** Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

33651 **ASYNCHRONOUS EVENTS**  
33652       Default.

33653 **STDOUT**  
33654       The input files shall be written to standard output, with the editing commands specified in the  
33655       script applied. If the **-n** option is specified, only those input lines selected by the script shall be  
33656       written to standard output.

33657 **STDERR**  
33658       Used only for diagnostic messages.

33659 **OUTPUT FILES**  
33660       The output files shall be text files whose formats are dependent on the editing commands given.

33661 **EXTENDED DESCRIPTION**  
33662       The *script* shall consist of editing commands of the following form:

33663       [*address* [ , *address* ] ] *function*

33664       where *function* represents a single-character command verb from the list in **Editing Commands**  
33665       in **sed** on page 881, followed by any applicable arguments.

33666       Zero or more <blank>s shall be accepted before the first address and before function. Any  
33667       number of semicolons shall be accepted before the first address.

33668       In default operation, *sed* cyclically shall copy a line of input, less its terminating <newline>, into  
33669       a pattern space (unless there is something left after a **D** command), apply in sequence all  
33670       commands whose addresses select that pattern space, and at the end of the script copy the  
33671       pattern space to standard output (except when **-n** is specified) and delete the pattern space.  
33672       Whenever the pattern space is written to standard output or a named file, *sed* shall immediately  
33673       follow it with a <newline>.

33674       Some of the editing commands use a hold space to save all or part of the pattern space for  
33675       subsequent retrieval. The pattern and hold spaces shall each be able to hold at least 8 192 bytes.

### 33676 Addresses in sed

33677 An address is either a decimal number that counts input lines cumulatively across files, a '\$' character that addresses the last line of input, or a context address (which consists of a BRE, as described in **Regular Expressions in sed**, preceded and followed by a delimiter, usually a slash).

33680 An editing command with no addresses shall select every pattern space.

33681 An editing command with one address shall select each pattern space that matches the address.

33682 An editing command with two addresses shall select the inclusive range from the first pattern space that matches the first address through the next pattern space that matches the second. (If the second address is a number less than or equal to the line number first selected, only one line shall be selected.) Starting at the first line following the selected range, *sed* shall look again for the first address. Thereafter, the process shall be repeated. Omitting either or both of the address components in the following form produces undefined results:

33688 [*address* [, *address* ] ]

### 33689 Regular Expressions in sed

33690 The *sed* utility shall support the BREs described in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 9.3, Basic Regular Expressions, with the following additions:

- 33692 • In a context address, the construction "`\cBREc`", where *c* is any character other than  
33693 backslash or <newline>, shall be identical to "`/BRE/`". If the character designated by *c*  
33694 appears following a backslash, then it shall be considered to be that literal character, which  
33695 shall not terminate the BRE. For example, in the context address "`.mc |`" the second *x* stands  
33696 for itself, so that the BRE is "`abcxdef`".
- 33697 • The escape sequence '`\n`' shall match a <newline> embedded in the pattern space. A literal  
33698 <newline> character shall not be used in the BRE of a context address or in the substitute  
33699 function.
- 33700 • If an RE is empty (that is, no pattern is specified) *sed* shall behave as if the last RE used in the  
33701 last command applied (either as an address or as part of a substitute command) was  
33702 specified.

### 33703 Editing Commands in sed

33704 In the following list of editing commands, the maximum number of permissible addresses for  
33705 each function is indicated by [*0addr*], [*1addr*], or [*2addr*], representing zero, one, or two  
33706 addresses.

33707 The argument *text* shall consist of one or more lines. Each embedded <newline> in the text shall  
33708 be preceded by a backslash. Other backslashes in text shall be removed, and the following  
33709 character shall be treated literally.

33710 The **r** and **w** command verbs, and the **w** flag to the **s** command, take an optional *rfile* (or *wfile*)  
33711 parameter, separated from the command verb letter or flag by one or more <blank>s;  
33712 implementations may allow zero separation as an extension.

33713 The argument *rfile* or the argument *wfile* shall terminate the editing command. Each *wfile* shall be  
33714 created before processing begins. Implementations shall support at least ten *wfile* arguments in  
33715 the script; the actual number (greater than or equal to 10) that shall be supported by the  
33716 implementation is unspecified. The use of the *wfile* parameter shall cause that file to be initially  
33717 created, if it does not exist, or shall replace the contents of an existing file.

33718 The **b**, **r**, **s**, **t**, **w**, **y**, and **:** command verbs shall accept additional arguments. The following  
33719 synopses indicate which arguments shall be separated from the command verbs by a single  
33720 <space>.

33721 The **a** and **r** commands schedule text for later output. The text specified for the **a** command, and  
33722 the contents of the file specified for the **r** command, shall be written to standard output just  
33723 before the next attempt to fetch a line of input when executing the **N** or **n** commands, or when  
33724 reaching the end of the script. If written when reaching the end of the script, and the **-n** option  
33725 was not specified, the text shall be written after copying the pattern space to standard output.  
33726 The contents of the file specified for the **r** command shall be as of the time the output is written,  
33727 not the time the **r** command is applied. The text shall be output in the order in which the **a** and **r**  
33728 commands were applied to the input.

33729 Command verbs other than **{**, **a**, **b**, **c**, **i**, **r**, **t**, **w**, **:**, and **#** can be followed by a semicolon, optional  
33730 <blank>s, and another command verb. However, when the **s** command verb is used with the **w**  
33731 flag, following it with another command in this manner produces undefined results.

33732 A function can be preceded by one or more '!' characters, in which case the function shall be  
33733 applied if the addresses do not select the pattern space. Zero or more <blank>s shall be accepted  
33734 before the first '!' character. It is unspecified whether <blank> characters can follow a '!'  
33735 character, and conforming applications shall not follow a '!' character with <blank>s.

33736 [2addr] {function  
33737 function  
33738 ...  
33739 } Execute a list of *sed* functions only when the pattern space is selected. The list of  
33740 *sed* functions shall be surrounded by braces and separated by <newline>s, as  
33741 follows. The braces can be preceded or followed by <blank>s. The functions can  
33742 be preceded by <blank>s, but shall not be followed by <blank>s. The <right-  
33743 brace> shall be preceded by a <newline> and can be preceded or followed by  
33744 <blank>s.

33745 [1addr]a\  
33746 text Write text to standard output as described previously.

33747 [2addr]b [label]  
33748 Branch to the **:** function bearing the *label*. If *label* is not specified, branch to the end  
33749 of the script. The implementation shall support *labels* recognized as unique up to  
33750 at least 8 characters; the actual length (greater than or equal to 8) that shall be  
33751 supported by the implementation is unspecified. It is unspecified whether  
33752 exceeding a label length causes an error or a silent truncation.

33753 [2addr]c\  
33754 text Delete the pattern space. With a 0 or 1 address or at the end of a 2-address range,  
33755 place *text* on the output and start the next cycle.

33756 [2addr]d Delete the pattern space and start the next cycle.

33757 [2addr]D Delete the initial segment of the pattern space through the first <newline> and  
33758 start the next cycle.

33759 [2addr]g Replace the contents of the pattern space by the contents of the hold space.

33760 [2addr]G Append to the pattern space a <newline> followed by the contents of the hold  
33761 space.

33762 [2addr]h Replace the contents of the hold space with the contents of the pattern space.

|       |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 33763 | [2addr]H                       | Append to the hold space a <newline> followed by the contents of the pattern space.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 33764 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33765 | [1addr]i\                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33766 | text                           | Write <i>text</i> to standard output.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33767 | [2addr]l                       | (The letter ell.) Write the pattern space to standard output in a visually unambiguous form. The characters listed in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Table 5-1, Escape Sequences and Associated Actions ('\\', '\a', '\b', '\f', '\r', '\t', '\v') shall be written as the corresponding escape sequence; the '\n' in that table is not applicable. Non-printable characters not in that table shall be written as one three-digit octal number (with a preceding backslash) for each byte in the character (most significant byte first). If the size of a byte on the system is greater than 9 bits, the format used for non-printable characters is implementation-dependent. |
| 33768 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33769 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33770 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33771 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33772 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33773 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33774 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33775 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33776 |                                | Long lines shall be folded, with the point of folding indicated by writing a backslash followed by a <newline>; the length at which folding occurs is unspecified, but should be appropriate for the output device. The end of each line shall be marked with a '\$'.                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 33777 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33778 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33779 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33780 | [2addr]n                       | Write the pattern space to standard output if the default output has not been suppressed, and replace the pattern space with the next line of input.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 33781 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33782 |                                | If no next line of input is available, the <b>n</b> command verb shall branch to the end of the script and quit without starting a new cycle.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 33783 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33784 | [2addr]N                       | Append the next line of input to the pattern space, using an embedded <newline> character to separate the appended material from the original material. Note that the current line number changes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 33785 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33786 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33787 |                                | If no next line of input is available, the <b>N</b> command verb shall branch to the end of the script and quit without starting a new cycle or copying the pattern space to standard output.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 33788 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33789 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33790 | [2addr]p                       | Write the pattern space to standard output.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 33791 | [2addr]P                       | Write the pattern space, up to the first <newline>, to standard output.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 33792 | [1addr]q                       | Branch to the end of the script and quit without starting a new cycle.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 33793 | [1addr]r rfile                 | Copy the contents of <i>rfile</i> to standard output as described previously. If <i>rfile</i> does not exist or cannot be read, it shall be treated as if it were an empty file, causing no error condition.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 33794 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33795 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33796 | [2addr]s/BRE/replacement/flags |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33797 |                                | Substitute the replacement string for instances of the BRE in the pattern space. Any character other than backslash or <newline> can be used instead of a slash to delimit the BRE and the replacement. Within the BRE and the replacement, the BRE delimiter itself can be used as a literal character if it is preceded by a backslash.                                                                                                                                                                                                                                                                                                                                                                           |
| 33798 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33799 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33800 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33801 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33802 |                                | An ampersand ('&') appearing in the replacement shall be replaced by the string matching the BRE. The special meaning of '&' in this context can be suppressed by preceding it by a backslash. The characters '\n', where <i>n</i> is a digit, shall be replaced by the text matched by the corresponding backreference expression. For each backslash ('\') encountered in scanning <i>replacement</i> from beginning to end, the backslash shall be discarded and the following character shall lose its special meaning (if any). It is unspecified what special meaning is given to any character                                                                                                               |
| 33803 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33804 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33805 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33806 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33807 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 33808 |                                |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

|       |                                  |                                                                                                                 |  |
|-------|----------------------------------|-----------------------------------------------------------------------------------------------------------------|--|
| 33809 |                                  | other than '&', '\', or digits.                                                                                 |  |
| 33810 |                                  | A line can be split by substituting a <newline> character into it. The application                              |  |
| 33811 |                                  | shall escape the <newline> in the replacement by preceding it by a backslash. A                                 |  |
| 33812 |                                  | substitution shall be considered to have been performed even if the replacement                                 |  |
| 33813 |                                  | string is identical to the string that it replaces. Any backslash used to alter the                             |  |
| 33814 |                                  | default meaning of a subsequent character shall be discarded from the BRE or the                                |  |
| 33815 |                                  | replacement before evaluating the BRE or using the replacement.                                                 |  |
| 33816 |                                  | The value of <i>flags</i> shall be zero or more of:                                                             |  |
| 33817 | <b>n</b>                         | Substitute for the <i>n</i> th occurrence only of the BRE found within the                                      |  |
| 33818 |                                  | pattern space.                                                                                                  |  |
| 33819 | <b>g</b>                         | Globally substitute for all non-overlapping instances of the BRE                                                |  |
| 33820 |                                  | rather than just the first one. If both <b>g</b> and <b>n</b> are specified, the results                        |  |
| 33821 |                                  | are unspecified.                                                                                                |  |
| 33822 | <b>p</b>                         | Write the pattern space to standard output if a replacement was                                                 |  |
| 33823 |                                  | made.                                                                                                           |  |
| 33824 | <b>w wfile</b>                   | Write. Append the pattern space to <i>wfile</i> if a replacement was made.                                      |  |
| 33825 |                                  | A conforming application shall precede the <i>wfile</i> argument with one                                       |  |
| 33826 |                                  | or more <blank>s. If the <b>w</b> flag is not the last flag value given in a                                    |  |
| 33827 |                                  | concatenation of multiple flag values, the results are undefined.                                               |  |
| 33828 | <b>[2addr]t [label]</b>          |                                                                                                                 |  |
| 33829 |                                  | Test. Branch to the <b>:</b> command verb bearing the <i>label</i> if any substitutions have been               |  |
| 33830 |                                  | made since the most recent reading of an input line or execution of a <b>t</b> . If <i>label</i> is             |  |
| 33831 |                                  | not specified, branch to the end of the script.                                                                 |  |
| 33832 | <b>[2addr]w wfile</b>            |                                                                                                                 |  |
| 33833 |                                  | Append {write) the pattern space to <i>wfile</i> .                                                              |  |
| 33834 | <b>[2addr]x</b>                  | Exchange the contents of the pattern and hold spaces.                                                           |  |
| 33835 | <b>[2addr]y/string1/string2/</b> |                                                                                                                 |  |
| 33836 |                                  | Replace all occurrences of characters in <i>string1</i> with the corresponding characters                       |  |
| 33837 |                                  | in <i>string2</i> . If a backslash followed by an <i>n</i> appear in <i>string1</i> or <i>string2</i> , the two |  |
| 33838 |                                  | characters shall be handled as a single <newline> character. If the number of                                   |  |
| 33839 |                                  | characters in <i>string1</i> and <i>string2</i> are not equal, or if any of the characters in <i>string1</i>    |  |
| 33840 |                                  | appear more than once, the results are undefined. Any character other than                                      |  |
| 33841 |                                  | backslash or <newline> can be used instead of slash to delimit the strings. If the                              |  |
| 33842 |                                  | delimiter is not <i>n</i> , within <i>string1</i> and <i>string2</i> , the delimiter itself can be used as a    |  |
| 33843 |                                  | literal character if it is preceded by a backslash. If a backslash character is                                 |  |
| 33844 |                                  | immediately followed by a backslash character in <i>string1</i> or <i>string2</i> , the two                     |  |
| 33845 |                                  | backslash characters shall be counted as a single literal backslash character. The                              |  |
| 33846 |                                  | meaning of a backslash followed by any character that is not <i>n</i> , a backslash, or the                     |  |
| 33847 |                                  | delimiter character is undefined.                                                                               |  |
| 33848 | <b>[0addr]:label</b>             | Do nothing. This command bears a <i>label</i> to which the <b>b</b> and <b>t</b> commands branch.               |  |
| 33849 | <b>[1addr]=</b>                  | Write the following to standard output:                                                                         |  |
| 33850 |                                  | "%d\n", <current line number>                                                                                   |  |
| 33851 | <b>[0addr]</b>                   | Ignore this empty command.                                                                                      |  |
| 33852 | <b>[0addr]#</b>                  | Ignore the '#' and the remainder of the line (treat them as a comment), with the                                |  |
| 33853 |                                  | single exception that if the first two characters in the script are "#n", the default                           |  |



33854                   output shall be suppressed; this shall be the equivalent of specifying **-n** on the  
33855                   command line.

### 33856 **EXIT STATUS**

33857           The following exit values shall be returned:

33858           **0**   Successful completion.

33859           **>0**   An error occurred.

### 33860 **CONSEQUENCES OF ERRORS**

33861           Default.

### 33862 **APPLICATION USAGE**

33863           Regular expressions match entire strings, not just individual lines, but a <newline> character is  
33864           matched by '**\n**' in a *sed* RE; a <newline> character is not allowed in an RE. Also note that '**\n**'  
33865           cannot be used to match a <newline> character at the end of an arbitrary input line; <newline>  
33866           characters appear in the pattern space as a result of the **N** editing command.

### 33867 **EXAMPLES**

33868           This *sed* script simulates the BSD *cat -s* command, squeezing excess blank lines from standard  
33869           input.

```
33870 sed -n '
33871 # Write non-empty lines.
33872 ./ {
33873 p
33874 d
33875 }
33876 # Write a single empty line, then look for more empty lines.
33877 /^$/ p
33878 # Get next line, discard the held <newline> (empty line),
33879 # and look for more empty lines.
33880 :Empty
33881 /^$/ {
33882 N
33883 s/./ /
33884 b Empty
33885 }
33886 # Write the non-empty line before going back to search
33887 # for the first in a set of empty lines.
33888 p
33889 '
```

### 33890 **RATIONALE**

33891           This volume of IEEE Std. 1003.1-200x requires implementations to support at least ten distinct  
33892           *wfiles*, matching historical practice on many implementations. Implementations are encouraged  
33893           to support more, but portable applications should not exceed this limit.

33894           The exit status codes specified here are different from those in System V. System V returns 2 for  
33895           garbled *sed* commands, but returns zero with its usage message or if the input file could not be  
33896           opened. The standard developers considered this to be a bug.

33897           The manner in which the **l** command writes non-printable characters was changed to avoid the  
33898           historical backspace-overstrike method, and other requirements to achieve unambiguous output  
33899           were added. See the RATIONALE for *ed* on page 369 for details of the format chosen, which is  
33900           the same as that chosen for *sed*.

33901 This volume of IEEE Std. 1003.1-200x requires implementations to provide pattern and hold  
 33902 spaces of at least 8 192 bytes, larger than the 4 000 bytes spaces used by some historical  
 33903 implementations, but less than the 20 480 bytes limit used in an early proposal. Implementations  
 33904 are encouraged to allocate dynamically larger pattern and hold spaces as needed.

33905 The requirements for acceptance of <blank>s and <space>s in command lines has been made  
 33906 more explicit than in early proposals to describe clearly the historical practice and to remove  
 33907 confusion about the phrase “protect initial blanks [*sic*] and tabs from the stripping that is done  
 33908 on every script line” that appears in much of the historical documentation of the *sed* utility  
 33909 description of text. (Not all implementations are known to have stripped <blank>s from text  
 33910 lines, although they all have allowed leading <blank>s preceding the address on a command  
 33911 line.)

33912 The treatment of ‘#’ comments differs from the SVID which only allows a comment as the first  
 33913 line of the script, but matches BSD-derived implementations. The comment character is treated  
 33914 as a command, and it has the same properties in terms of being accepted with leading <blank>s;  
 33915 the BSD implementation has historically supported this.

33916 Early proposals required that a *script\_file* have at least one non-comment line. Some historical  
 33917 implementations have behaved in unexpected ways if this were not the case. The standard  
 33918 developers considered that this was incorrect behavior and that application developers should  
 33919 not have to avoid this feature. A correct implementation of this volume of IEEE Std. 1003.1-200x  
 33920 shall permit *script\_files* that consist only of comment lines.

33921 Early proposals indicated that if **-e** and **-f** options were intermixed, all **-e** options were  
 33922 processed before any **-f** options. This has been changed to process them in the order presented  
 33923 because it matches historical practice and is more intuitive.

33924 The treatment of the **p** flag to the **s** command differs between System V and BSD-based systems  
 33925 when the default output is suppressed. In the two examples:

```
33926 echo a | sed 's/a/A/p'
33927 echo a | sed -n 's/a/A/p'
```

33928 This volume of IEEE Std. 1003.1-200x, BSD, System V documentation, and the SVID indicate that  
 33929 the first example should write two lines with **A**, whereas the second should write one. Some  
 33930 System V systems write the **A** only once in both examples because the **p** flag is ignored if the **-n**  
 33931 option is not specified.

33932 This is a case of a diametrical difference between systems that could not be reconciled through  
 33933 the compromise of declaring the behavior to be unspecified. The SVID/BSD/System V  
 33934 documentation behavior was adopted for this volume of IEEE Std. 1003.1-200x because:

- 33935 • No known documentation for any historic system describes the interaction between the **p**  
 33936 flag and the **-n** option.
- 33937 • The selected behavior is more correct as there is no technical justification for any interaction  
 33938 between the **p** flag and the **-n** option. A relationship between **-n** and the **p** flag might imply  
 33939 that they are only used together, but this ignores valid scripts that interrupt the cyclical  
 33940 nature of the processing through the use of the **D**, **d**, **q**, or branching commands. Such scripts  
 33941 rely on the **p** suffix to write the pattern space because they do not make use of the default  
 33942 output at the “bottom” of the script.
- 33943 • Because the **-n** option makes the **p** flag unnecessary, any interaction would only be useful if  
 33944 *sed* scripts were written to run both with and without the **-n** option. This is believed to be  
 33945 unlikely. It is even more unlikely that programmers have coded the **p** flag expecting it to be  
 33946 unnecessary. Because the interaction was not documented, the likelihood of a programmer  
 33947 discovering the interaction and depending on it is further decreased.

- 33948 • Finally, scripts that break under the specified behavior produce too much output instead of
- 33949 too little, which is easier to diagnose and correct.
- 33950 The form of the substitute command that uses the **n** suffix was limited to the first 512 matches in
- 33951 an early proposal. This limit has been removed because there is no reason an editor processing
- 33952 lines of {LINE\_MAX} length should have this restriction. The command **s/a/A/2047** should be
- 33953 able to substitute the 2047th occurrence of **a** on a line.
- 33954 The **b**, **t**, and **:** commands are documented to ignore leading white space, but no mention is
- 33955 made of trailing white space. Historical implementations of *sed* assigned different locations to
- 33956 the labels '**x**' and "**x**". This is not useful, and leads to subtle programming errors, but it is
- 33957 historical practice, and changing it could theoretically break working scripts. Implementors are
- 33958 encouraged to provide warning messages about labels that are never used or jumps to labels
- 33959 that do not exist.
- 33960 Historically, the *sed* **!** and **}** editing commands did not permit multiple commands on a single
- 33961 line using a semicolon as a command delimiter. Implementations are permitted, but not
- 33962 required, to support this extension.
- 33963 **FUTURE DIRECTIONS**
- 33964 None.
- 33965 **SEE ALSO**
- 33966 *awk*, *ed*, *grep*
- 33967 **CHANGE HISTORY**
- 33968 First released in Issue 2.
- 33969 **Issue 4**
- 33970 Aligned with the ISO/IEC 9945-2:1993 standard.
- 33971 **Issue 5**
- 33972 FUTURE DIRECTIONS section added.
- 33973 **Issue 6**
- 33974 The following new requirements on POSIX implementations derive from alignment with the
- 33975 Single UNIX Specification:
- 33976 • Implementations are required to support at least ten *wfile* arguments in an editing command.
- 33977 The EXTENDED DESCRIPTION is changed to align with the IEEE P1003.2b draft standard.

## 33978 NAME

33979 sh — shell, the standard command language interpreter

## 33980 SYNOPSIS

33981 MAN sh [-abCefhimnuvx][-o option][+abCefhimnuvx][+o option]  
33982 [command\_file [argument...]]

33983 MAN sh -c[-abCefhimnuvx][-o option][+abCefhimnuvx][+o option]command\_string  
33984 [command\_name [argument...]]

33985 MAN sh -s[-abCefhimnuvx][-o option][+abCefhimnuvx][+o option][argument]

## 33986 DESCRIPTION

33987 The *sh* utility is a command language interpreter that shall execute commands read from a  
33988 command line string, the standard input, or a specified file. The application shall ensure that the  
33989 commands to be executed are expressed in the language described in Chapter 2 on page 35.

33990 MAN Path name expansion does not fail due to the size of a file.

33991 Shell input and output redirections have an implementation-dependent offset maximum that is  
33992 established in the open file description.

## 33993 OPTIONS

33994 The *sh* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
33995 Section 12.2, Utility Syntax Guidelines.

33996 The **-a**, **-b**, **-C**, **-e**, **-f**, **-m**, **-n**, **-o option**, **-u**, **-v**, and **-x** options are described as part of the *set*  
33997 MAN utility in Section 2.14 on page 96. The option letters derived from the *set* special built-in shall  
33998 also be accepted with a leading plus sign ('+') instead of a leading hyphen (meaning the reverse  
33999 case of the option as described in this volume of IEEE Std. 1003.1-200x).

34000 The following additional options shall be supported:

34001 **-c** Read commands from the *command\_string* operand. Set the value of special  
34002 parameter 0 (see Section 2.5.2 on page 43) from the value of the *command\_name*  
34003 operand and the positional parameters (\$1, \$2, and so on) in sequence from the  
34004 remaining *argument* operands. No commands shall be read from the standard  
34005 input.

34006 **-i** Specify that the shell is *interactive*; see below. An implementation may treat  
34007 specifying the **-i** option as an error if the real user ID of the calling process does  
34008 not equal the effective user ID or if the real group ID does not equal the effective  
34009 group ID.

34010 **-s** Read commands from the standard input.

34011 If there are no operands and the **-c** option is not specified, the **-s** option shall be assumed.

34012 If the **-i** option is present, or if there are no operands and the shell's standard input and standard  
34013 error are attached to a terminal, the shell is considered to be *interactive*.

## 34014 OPERANDS

34015 The following operands shall be supported:

34016 **-** A single hyphen is treated as the first operand and then ignored. If both '-' and  
34017 "—" are given as arguments, or if other operands precede the single hyphen, the  
34018 results are undefined.

34019 *argument* The positional parameters (\$1, \$2, and so on) shall be set to *arguments*, if any.

34020 *command\_file* The path name of a file containing commands. If the path name contains one or  
 34021 more slash characters, the implementation attempts to read that file; the file need  
 34022 not be executable. If the path name does not contain a slash character:

- 34023 • The implementation shall attempt to read that file from the current working  
 34024 directory; the file need not be executable.
- 34025 • If the file is not in the current working directory, the implementation may  
 34026 perform a search for an executable file using the value of *PATH*, as described in  
 34027 Section 2.9.1.1 on page 69.

34028 Special parameter 0 (see Section 2.5.2 on page 43) shall be set to the value of  
 34029 *command\_file*. If *sh* is called using a synopsis form that omits *command\_file*, special  
 34030 parameter 0 shall be set to the value of the first argument passed to *sh* from its  
 34031 parent (for example, *argv*[0] for a C program), which is normally a path name used  
 34032 to execute the *sh* utility.

34033 *command\_name*

34034 A string assigned to special parameter 0 when executing the commands in  
 34035 *command\_string*. If *command\_name* is not specified, special parameter 0 shall be set  
 34036 to the value of the first argument passed to *sh* from its parent (for example, *argv*[0]  
 34037 for a C program), which is normally a path name used to execute the *sh* utility.

34038 *command\_string*

34039 A string that shall be interpreted by the shell as one or more commands, as if the  
 34040 string were the argument to the *system()* function defined in the System Interfaces  
 34041 volume of IEEE Std. 1003.1-200x. If the *command\_string* operand is an empty string,  
 34042 *sh* shall exit with a zero exit status.

## 34043 STDIN

34044 The standard input shall be used only if one of the following is true:

- 34045 • The *-s* option is specified.
- 34046 • The *-c* option is not specified and no operands are specified.
- 34047 • The script executes one or more commands that require input from standard input (such as a  
 34048 *read* command that does not redirect its input).

34049 See the INPUT FILES section.

34050 When the shell is using standard input and it invokes a command that also uses standard input,  
 34051 the shell shall ensure that the standard input file pointer points directly after the command it has  
 34052 read when the command begins execution. It shall not read ahead in such a manner that any  
 34053 characters intended to be read by the invoked command are consumed by the shell (whether  
 34054 interpreted by the shell or not) or that characters that are not read by the invoked command are  
 34055 not seen by the shell. When the command expecting to read standard input is started  
 34056 asynchronously by an interactive shell, it is unspecified whether characters are read by the  
 34057 command or interpreted by the shell.

34058 If the standard input to *sh* is a FIFO or terminal device and is set to non-blocking reads, then *sh*  
 34059 shall enable blocking reads on standard input. This shall remain in effect when the command  
 34060 completes.

## 34061 INPUT FILES

34062 The input file shall be a text file, except that line lengths shall be unlimited. If the input file is  
 34063 empty or consists solely of blank lines or comments, or both, *sh* shall exit with a zero exit status.

## 34064 ENVIRONMENT VARIABLES

34065 The following environment variables shall affect the execution of *sh*:

34066 *ENV* This variable, when and only when an interactive shell is invoked, shall be  
 34067 subjected to parameter expansion (see Section 2.6.2 on page 51) by the shell, and  
 34068 the resulting value shall be used as a path name of a file containing shell  
 34069 commands to execute in the current environment. The file need not be executable.  
 34070 If the expanded value of *ENV* is not an absolute path name, the results are  
 34071 unspecified. *ENV* shall be ignored if the real and effective user IDs or real and  
 34072 effective group IDs of the user are different.

34073 *FCEDIT* This variable, when expanded by the shell, determines the default value for the *-e*  
 34074 *editor* option's *editor* option-argument. If *FCEDIT* is null or unset, *ed* shall be used  
 34075 as the editor. This volume of IEEE Std. 1003.1-200x specifies the effects of this  
 34076 variable only for systems supporting the User Portability Utilities option.

34077 *HISTFILE* Determine a path name naming a command history file. If the *HISTFILE* variable is  
 34078 not set, the shell may attempt to access or create a file *.sh\_history* in the directory  
 34079 referred to by the *HOME* environment variable. If the shell cannot obtain both read  
 34080 and write access to, or create, the history file, it shall use an unspecified  
 34081 mechanism that allows the history to operate properly. (References to history  
 34082 "file" in this section shall be understood to mean this unspecified mechanism in  
 34083 such cases.) An implementation may choose to access this variable only when  
 34084 initializing the history file; this initialization shall occur when *fc* or *sh* first attempt  
 34085 to retrieve entries from, or add entries to, the file, as the result of commands issued  
 34086 by the user, the file named by the *ENV* variable, or implementation-dependent  
 34087 system start-up files. (The initialization process for the history file can be  
 34088 dependent on the system start-up files, in that they may contain commands that  
 34089 effectively preempt the user's settings of *HISTFILE* and *HISTSIZE*. For example,  
 34090 function definition commands are recorded in the history file, unless the *set -o*  
 34091 *nolog* option is set. If the system administrator includes function definitions in  
 34092 some system start-up file called before the *ENV* file, the history file is initialized  
 34093 before the user gets a chance to influence its characteristics.) In some historical  
 34094 shells, the history file is initialized just after the *ENV* file has been processed.  
 34095 Therefore, it is implementation-dependent whether changes made to *HISTFILE*  
 34096 after the history file has been initialized are effective. Implementations may  
 34097 choose to disable the history list mechanism for users with appropriate privileges  
 34098 who do not set *HISTFILE*; the specific circumstances under which this occurs are  
 34099 implementation-dependent. If more than one instance of the shell is using the  
 34100 same history file, it is unspecified how updates to the history file from those shells  
 34101 interact. As entries are deleted from the history file, they shall be deleted oldest  
 34102 first. It is unspecified when history file entries are physically removed from the  
 34103 history file. This volume of IEEE Std. 1003.1-200x specifies the effects of this  
 34104 variable only for systems supporting the User Portability Utilities option.

34105 *HISTSIZE* Determine a decimal number representing the limit to the number of previous  
 34106 commands that are accessible. If this variable is unset, an unspecified default  
 34107 greater than or equal to 128 shall be used. The maximum number of commands in  
 34108 the history list is unspecified, but shall be at least 128. An implementation may  
 34109 choose to access this variable only when initializing the history file, as described  
 34110 under *HISTFILE*. Therefore, it is unspecified whether changes made to *HISTSIZE*  
 34111 after the history file has been initialized are effective.

34112 *HOME* Determine the path name of the user's home directory. The contents of *HOME* are  
 34113 used in Tilde Expansion as described in Section 2.6.1 on page 50. This volume of

|       |                    |                                                                                                            |
|-------|--------------------|------------------------------------------------------------------------------------------------------------|
| 34114 |                    | IEEE Std. 1003.1-200x specifies the effects of this variable only for systems                              |
| 34115 |                    | supporting the User Portability Utilities option.                                                          |
| 34116 | <i>IFS</i>         | <i>Input field separators</i> : a string treated as a list of characters that shall be used for            |
| 34117 |                    | field splitting and to split lines into words with the <i>read</i> command. See Section                    |
| 34118 |                    | 2.6.5 on page 58. If <i>IFS</i> is not set, the shell shall behave as if the value of <i>IFS</i> were      |
| 34119 |                    | the <space>, <tab>, and <newline> characters. Implementations may ignore the                               |
| 34120 |                    | value of <i>IFS</i> in the environment at the time <i>sh</i> is invoked, treating <i>IFS</i> as if it were |
| 34121 |                    | not set.                                                                                                   |
| 34122 | <i>LANG</i>        | Provide a default value for the internationalization variables that are unset or null.                     |
| 34123 |                    | If <i>LANG</i> is unset or null, the corresponding value from the implementation-                          |
| 34124 |                    | dependent default locale shall be used. If any of the internationalization variables                       |
| 34125 |                    | contains an invalid setting, the utility shall behave as if none of the variables had                      |
| 34126 |                    | been defined.                                                                                              |
| 34127 | <i>LC_ALL</i>      | If set to a non-empty string value, override the values of all the other                                   |
| 34128 |                    | internationalization variables.                                                                            |
| 34129 | <i>LC_COLLATE</i>  |                                                                                                            |
| 34130 |                    | Determine the behavior of range expressions, equivalence classes and multi-                                |
| 34131 |                    | character collating elements within pattern matching.                                                      |
| 34132 | <i>LC_CTYPE</i>    | Determine the locale for the interpretation of sequences of bytes of text data as                          |
| 34133 |                    | characters (for example, single-byte as opposed to multi-byte characters in                                |
| 34134 |                    | arguments and input files), which characters are defined as letters (character class                       |
| 34135 |                    | <b>alpha</b> ), and the behavior of character classes within pattern matching.                             |
| 34136 | <i>LC_MESSAGES</i> |                                                                                                            |
| 34137 |                    | Determine the locale that should be used to affect the format and contents of                              |
| 34138 |                    | diagnostic messages written to standard error.                                                             |
| 34139 | <i>MAIL</i>        | Determine a path name of the user's mailbox file for purposes of incoming mail                             |
| 34140 |                    | notification. If this variable is set, the shell shall inform the user if the file named by                |
| 34141 |                    | the variable is created or if its modification time has changed. Informing the user                        |
| 34142 |                    | shall be accomplished by writing a string of unspecified format to standard error                          |
| 34143 |                    | prior to the writing of the next primary prompt string after the completion of an                          |
| 34144 |                    | interval defined by the <i>MAILCHECK</i> variable. The user shall be informed only if                      |
| 34145 |                    | <i>MAIL</i> is set and <i>MAILPATH</i> is not set. This volume of IEEE Std. 1003.1-200x                    |
| 34146 |                    | specifies the effects of this variable only for systems supporting the User                                |
| 34147 |                    | Portability Utilities option.                                                                              |
| 34148 | <i>MAILCHECK</i>   |                                                                                                            |
| 34149 |                    | Establish a decimal integer value that specifies how often (in seconds) the shell                          |
| 34150 |                    | shall check for the arrival of mail in the files specified by the <i>MAILPATH</i> or <i>MAIL</i>           |
| 34151 |                    | variables. The default value shall be 600 seconds. If set to zero, the shell shall check                   |
| 34152 |                    | before issuing each primary prompt. This volume of IEEE Std. 1003.1-200x                                   |
| 34153 |                    | specifies the effects of this variable only for systems supporting the User                                |
| 34154 |                    | Portability Utilities option.                                                                              |
| 34155 | <i>MAILPATH</i>    | Provide a list of path names and optional messages separated by colons. If this                            |
| 34156 |                    | variable is set, the shell shall inform the user if any of the files named by the                          |
| 34157 |                    | variable are created or if any of their modification times change. (See the preceding                      |
| 34158 |                    | entry for <i>MAIL</i> for descriptions of mail arrival and user informing.) Each path                      |
| 34159 |                    | name can be followed by '%' and a string that shall be subjected to parameter                              |
| 34160 |                    | expansion and written to standard error when the modification time changes. If a                           |
| 34161 |                    | '%' character in the path name is preceded by a backslash, it shall be treated as a                        |

- 34162 literal ' % ' in the path name. The default message is unspecified.
- 34163 The *MAILPATH* environment variable takes precedence over the *MAIL* variable.
- 34164 This volume of IEEE Std. 1003.1-200x specifies the effects of this variable only for
- 34165 systems supporting the User Portability Utilities option.
- 34166 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.
- 34167 **PATH** Establish a string formatted as described in the System Interface Definitions
- 34168 volume of IEEE Std. 1003.1-200x, Chapter 8, Environment Variables, used to effect
- 34169 command interpretation; see Section 2.9.1.1 on page 69.
- 34170 **PWD** This variable shall represent an absolute path name of the current working
- 34171 directory. Assignments to this variable may be ignored unless the value is an
- 34172 absolute path name of the current working directory and there are no file name
- 34173 components of dot or dot-dot.
- 34174 **ASYNCHRONOUS EVENTS**
- 34175 Default.
- 34176 **STDOUT**
- 34177 See the **STDERR** section.
- 34178 **STDERR**
- 34179 Except as otherwise stated (by the descriptions of any invoked utilities or in interactive mode),
- 34180 standard error is used only for diagnostic messages.
- 34181 **OUTPUT FILES**
- 34182 None.
- 34183 **EXTENDED DESCRIPTION**
- 34184 See Chapter 2. The following additional capabilities are supported on systems supporting the
- 34185 User Portability Utilities option.
- 34186 **Command History List**
- 34187 When the *sh* utility is being used interactively, it shall maintain a list of commands previously
- 34188 entered from the terminal in the file named by the *HISTFILE* environment variable. The type,
- 34189 size, and internal format of this file are unspecified. Multiple *sh* processes can share access to the
- 34190 file for a user, if file access permissions allow this; see the description of the *HISTFILE*
- 34191 environment variable.
- 34192 **Command Line Editing**
- 34193 When *sh* is being used interactively from a terminal, the current command and the command
- 34194 history (see *fc* on page 470) can be edited using *vi*-mode command line editing. This mode uses
- 34195 commands, described below, similar to a subset of those described in the *vi* utility.
- 34196 Implementations may offer other command line editing modes corresponding to other editing
- 34197 utilities.
- 34198 The command *set -o vi* shall enable *vi*-mode editing and place *sh* into *vi* insert mode (see
- 34199 **Command Line Editing (vi-mode)** on page 893). This command also shall disable any other
- 34200 editing mode that the implementation may provide. The command *set +o vi* disables *vi*-mode
- 34201 editing.
- 34202 Certain block-mode terminals may be unable to support shell command line editing. If a
- 34203 terminal is unable to provide either edit mode, it need not be possible to *set -o vi* when using the
- 34204 shell on this terminal.



34205 In the following sections, the characters *erase*, *interrupt*, *kill*, and *end-of-file* are those set by the  
34206 *stty* utility.

### 34207 **Command Line Editing (vi-mode)**

34208 With *vi*-mode enabled, *sh* can be switched between insert mode and command mode.

34209 When in insert mode, an entered character shall be inserted into the command line, except as  
34210 noted in **vi Line Editing Insert Mode**. Upon entering *sh* and after termination of the previous  
34211 command, *sh* shall be in insert mode.

34212 Typing an escape character shall switch *sh* into command mode (see **vi Line Editing Command**  
34213 **Mode** on page 894). In command mode, an entered character shall either invoke a defined  
34214 operation, is used as part of a multi-character operation, or is treated as an error. A character that  
34215 is not recognized as part of an editing command shall terminate any specific editing command  
34216 and shall alert the terminal. Typing the *interrupt* character in command mode shall cause *sh* to  
34217 terminate command line editing on the current command line, reissue the prompt on the next  
34218 line of the terminal, and reset the command history (see *fc* on page 470) so that the most recently  
34219 executed command is the previous command (that is, the command that was being edited when  
34220 it was interrupted is not reentered into the history).

34221 In the following sections, the phrase “move the cursor to the beginning of the word” shall mean  
34222 “move the cursor to the first character of the current word” and the phrase “move the cursor to  
34223 the end of the word” shall mean “move the cursor to the last character of the current word”. The  
34224 phrase “beginning of the command line” indicates the point between the end of the prompt  
34225 string issued by the shell (or the beginning of the terminal line, if there is no prompt string) and  
34226 the first character of the command text.

### 34227 **vi Line Editing Insert Mode**

34228 While in insert mode, any character typed shall be inserted in the current command line, unless  
34229 it is from the following set.

34230 <newline> Execute the current command line being edited.

34231 *erase* Delete the character previous to the current cursor position and move the current  
34232 cursor position back one character. In insert mode, characters shall be erased from  
34233 both the screen and the buffer when backspacing.

34234 *interrupt* Terminate command line editing with the same effects as described for  
34235 interrupting command mode; see **Command Line Editing (vi-mode)**.

34236 *kill* Clear all the characters from the input line.

34237 <control>-V Insert the next character input, even if the character is otherwise a special insert  
34238 mode character.

34239 <control>-W Delete the characters from the one preceding the cursor to the preceding word  
34240 boundary. The word boundary in this case is the closer to the cursor of either the  
34241 beginning of the line or a character that is in neither the **blank** nor **punct** character  
34242 classification of the current locale.

34243 *end-of-file* Interpreted as the end of input in *sh*. This interpretation shall occur only at the  
34244 beginning of an input line. If *end-of-file* is entered other than at the beginning of the  
34245 line, the results are unspecified.

34246 <ESC> Place *sh* into command mode.

**vi Line Editing Command Mode**

In command mode for the command line editing feature, decimal digits not beginning with 0 that precede a command letter shall be remembered. Some commands use these decimal digits as a count number that affects the operation.

The term *motion command* represents one of the commands:

<space> 0 b F l W ^ \$ ; E f T w | , B e h t

Any command that modifies the current line shall cause a copy of the current line to be made at the end of the command history, the current line shall become that copy, and the edit is performed on that copy.

Any command that is preceded by *count* shall take a count (the numeric value of any preceding decimal digits). Unless otherwise noted, this count shall cause the specified operation to repeat by the number of times specified by the count. Also unless otherwise noted, a *count* that is out of range is considered an error condition and shall alert the terminal, but neither the cursor position, nor the command line, shall change.

The terms *word* and *bigword* are used as defined in the *vi* description. The term *save buffer* corresponds to the term *unnamed buffer* in *vi*.

The following commands shall be recognized in command mode:

<newline> Execute the current command line being edited.

<control>-L Redraw the current command line. Position the cursor at the same location on the new command line.

# Insert the character '#' at the beginning of the current command line and treat the current command line as a comment. This line shall be entered into the command history; see *fc* on page 470.

= Display the possible shell word expansions (see Section 2.6 on page 49) of the bigword at the current command line position. These expansions shall be displayed on subsequent terminal lines. If the bigword contains none of the characters '?', '\*', or '[', an asterisk('\*') shall be implicitly assumed at the end. If any directories are matched, these expansions shall have a '/' character appended. After the expansion, the line shall be redrawn, the cursor is repositioned at the current cursor position, and *sh* shall be placed in command mode.

\ Perform path name expansion (see Section 2.6.6 on page 59) on the current bigword, up to the largest set of characters that can be matched uniquely. If the bigword contains none of the characters '?', '\*', or '[', an asterisk('\*') shall be implicitly assumed at the end. This maximal expansion then shall replace the original bigword in the command line, and the cursor shall be placed after this expansion. If the resulting bigword completely and uniquely matches a directory, a '/' character shall be inserted directly after the bigword. If some other file is completely matched, a single <space> character shall be inserted after the bigword. After this operation, *sh* shall be placed in insert mode.

\* Perform path name expansion on the current bigword and insert all expansions into the command to replace the current bigword, with each expansion separated by a single <space> character. If at the end of the line, the current cursor position shall be moved to the first column position following the expansions and *sh* shall be placed in insert mode. Otherwise, the current cursor position shall be the last column position of the first character after the expansions and *sh* shall be placed in insert mode. If the current bigword contains none of the characters '?', '\*', or

|       |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 34293 |                             | ' [ ' , before the operation, an asterisk shall be implicitly assumed at the end.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 34294 | <b>@letter</b>              | Insert the value of the alias named <i>_letter</i> . The symbol <i>letter</i> represents a single alphabetic character from the portable character set; implementations may support additional characters as an extension. If the alias <i>_letter</i> contains other editing commands, these commands shall be performed as part of the insertion. If no alias <i>_letter</i> is enabled, this command shall have no effect.                                                                                                                                                                                                                                                                                                                       |
| 34295 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34296 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34297 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34298 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34299 | <b>[count]~</b>             | Convert, if the current character is a lowercase letter, to the equivalent uppercase letter and <i>vice versa</i> , as prescribed by the current locale. The current cursor position then shall be advanced by one character. If the cursor was positioned on the last character of the line, the case conversion shall occur, but the cursor shall not advance. If the ' ~ ' command is preceded by a <i>count</i> , that number of characters shall be converted, and the cursor shall be advanced to the character position after the last character converted. If the <i>count</i> is larger than the number of characters after the cursor, this shall not be considered an error; the cursor shall advance to the last character on the line. |
| 34300 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34301 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34302 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34303 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34304 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34305 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34306 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34307 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34308 | <b>[count].</b>             | Repeat the most recent non-motion command, even if it was executed on an earlier command line. If the previous command was preceded by a <i>count</i> , and no count is given on the ' . ' command, the count from the previous command shall be included as part of the repeated command. If the ' . ' command is preceded by a <i>count</i> , this shall override any <i>count</i> argument to the previous command. The <i>count</i> specified in the ' . ' command shall become the count for subsequent ' . ' commands issued without a count.                                                                                                                                                                                                 |
| 34309 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34310 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34311 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34312 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34313 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34314 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34315 | <b>[number]v</b>            | Invoke the <i>vi</i> editor to edit the current command line in a temporary file. When the editor exits, the commands in the temporary file shall be executed. If a <i>number</i> is prefixed to the command, it specifies the command number in the command history to be edited, rather than the current command line.                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 34316 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34317 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34318 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34319 | <b>[count]l</b> (ell)       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34320 | <b>[count]&lt;space&gt;</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34321 |                             | Move the current cursor position to the next character position. If the cursor was positioned on the last character of the line, the terminal shall be alerted and the cursor shall not be advanced. If the <i>count</i> is larger than the number of characters after the cursor, this shall not be considered an error; the cursor shall advance to the last character on the line.                                                                                                                                                                                                                                                                                                                                                               |
| 34322 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34323 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34324 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34325 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34326 | <b>[count]h</b>             | Move the current cursor position to the <i>count</i> th (default 1) previous character position. If the cursor was positioned on the first character of the line, the terminal shall be alerted and the cursor shall not be moved. If the count is larger than the number of characters before the cursor, this shall not be considered an error; the cursor shall move to the first character on the line.                                                                                                                                                                                                                                                                                                                                         |
| 34327 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34328 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34329 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34330 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34331 | <b>[count]w</b>             | Move to the start of the next word. If the cursor was positioned on the last character of the line, the terminal shall be alerted and the cursor shall not be advanced. If the <i>count</i> is larger than the number of words after the cursor, this shall not be considered an error; the cursor shall advance to the last character on the line.                                                                                                                                                                                                                                                                                                                                                                                                 |
| 34332 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34333 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34334 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34335 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34336 | <b>[count]W</b>             | Move to the start of the next bigword. If the cursor was positioned on the last character of the line, the terminal shall be alerted and the cursor shall not be advanced. If the <i>count</i> is larger than the number of bigwords after the cursor, this shall not be considered an error; the cursor shall advance to the last character on the line.                                                                                                                                                                                                                                                                                                                                                                                           |
| 34337 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34338 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34339 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34340 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

|       |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 34341 | <b>[count]e</b>  | Move to the end of the current word. If at the end of a word, move to the end of the next word. If the cursor was positioned on the last character of the line, the terminal shall be alerted and the cursor shall not be advanced. If the <i>count</i> is larger than the number of words after the cursor, this shall not be considered an error; the cursor shall advance to the last character on the line.                                     |
| 34342 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34343 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34344 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34345 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34346 | <b>[count]E</b>  | Move to the end of the current bigword. If at the end of a bigword, move to the end of the next bigword. If the cursor was positioned on the last character of the line, the terminal shall be alerted and the cursor shall not be advanced. If the <i>count</i> is larger than the number of bigwords after the cursor, this shall not be considered an error; the cursor shall advance to the last character on the line.                         |
| 34347 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34348 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34349 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34350 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34351 | <b>[count]b</b>  | Move to the beginning of the current word. If at the beginning of a word, move to the beginning of the previous word. If the cursor was positioned on the first character of the line, the terminal shall be alerted and the cursor shall not be moved. If the <i>count</i> is larger than the number of words preceding the cursor, this shall not be considered an error; the cursor shall return to the first character on the line.             |
| 34352 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34353 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34354 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34355 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34356 | <b>[count]B</b>  | Move to the beginning of the current bigword. If at the beginning of a bigword, move to the beginning of the previous bigword. If the cursor was positioned on the first character of the line, the terminal shall be alerted and the cursor shall not be moved. If the <i>count</i> is larger than the number of bigwords preceding the cursor, this shall not be considered an error; the cursor shall return to the first character on the line. |
| 34357 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34358 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34359 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34360 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34361 | <b>^</b>         | Move the current cursor position to the first character on the input line that is not a <blank> character.                                                                                                                                                                                                                                                                                                                                          |
| 34362 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34363 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34364 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34365 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34365 | <b>\$</b>        | Move to the last character position on the current command line.                                                                                                                                                                                                                                                                                                                                                                                    |
| 34366 | <b>0</b>         | (Zero.) Move to the first character position on the current command line.                                                                                                                                                                                                                                                                                                                                                                           |
| 34367 | <b>[count]  </b> | Move to the <i>count</i> th character position on the current command line. If no number is specified, move to the first position. The first character position shall be numbered 1. If the count is larger than the number of characters on the line, this shall not be considered an error; the cursor shall be placed on the last character on the line.                                                                                         |
| 34368 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34369 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34370 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34371 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34372 | <b>[count]fc</b> | Move to the first occurrence of the character 'c' that occurs after the current cursor position. If the cursor was positioned on the last character of the line, the terminal shall be alerted and the cursor shall not be advanced. If the character 'c' does not occur in the line after the current cursor position, the terminal shall be alerted and the cursor shall not be moved.                                                            |
| 34373 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34374 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34375 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34376 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34377 | <b>[count]Fc</b> | Move to the first occurrence of the character 'c' that occurs before the current cursor position. If the cursor was positioned on the first character of the line, the terminal shall be alerted and the cursor shall not be moved. If the character 'c' does not occur in the line before the current cursor position, the terminal shall be alerted and the cursor shall not be moved.                                                            |
| 34378 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34379 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34380 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34381 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34382 | <b>[count]tc</b> | Move to the character before the first occurrence of the character 'c' that occurs after the current cursor position. If the cursor was positioned on the last character of the line, the terminal shall be alerted and the cursor shall not be advanced. If the character 'c' does not occur in the line after the current cursor position, the terminal shall be alerted and the cursor shall not be moved.                                       |
| 34383 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34384 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34385 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34386 |                  |                                                                                                                                                                                                                                                                                                                                                                                                                                                     |

|       |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------|-----------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 34387 | <b>[count]Tc</b>      | Move to the character after the first occurrence of the character 'c' that occurs before the current cursor position. If the cursor was positioned on the first character of the line, the terminal shall be alerted and the cursor shall not be moved. If the character 'c' does not occur in the line before the current cursor position, the terminal shall be alerted and the cursor shall not be moved.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34388 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34389 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34390 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34391 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34392 | <b>[count];</b>       | Repeat the most recent <b>f</b> , <b>F</b> , <b>t</b> , or <b>T</b> command. Any number argument on that previous command shall be ignored. Errors are those described for the repeated command.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 34393 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34394 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34395 | <b>[count],</b>       | Repeat the most recent <b>f</b> , <b>F</b> , <b>t</b> , or <b>T</b> command. Any number argument on that previous command shall be ignored. However, reverse the direction of that command.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 34396 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34397 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34398 | <b>a</b>              | Enter insert mode after the current cursor position. Characters that are entered shall be inserted before the next character.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 34399 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34400 | <b>A</b>              | Enter insert mode after the end of the current command line.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34401 | <b>i</b>              | Enter insert mode at the current cursor position. Characters that are entered are inserted before the current character.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| 34402 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34403 | <b>I</b>              | Enter insert mode at the beginning of the current command line.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34404 | <b>R</b>              | Enter insert mode, replacing characters from the command line beginning at the current cursor position.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 34405 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34406 | <b>[count]cmotion</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34407 |                       | Delete the characters between the current cursor position and the cursor position that would result from the specified <i>motion</i> command. Then enter insert mode before the first character following any deleted characters. If <i>count</i> is specified, it shall be applied to the motion command. A <i>count</i> shall be ignored for the following motion commands:                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 34408 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34409 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34410 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34411 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34412 |                       | 0    ^    \$    c                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 34413 |                       | If the <i>motion</i> command is the character 'c', the current command line shall be cleared and insert mode shall be entered. If the <i>motion</i> command would move the current cursor position toward the beginning of the command line, the character under the current cursor position shall not be deleted. If the motion command would move the current cursor position toward the end of the command line, the character under the current cursor position shall be deleted. If the <i>count</i> is larger than the number of characters between the current cursor position and the end of the command line toward which the motion command would move the cursor, this shall not be considered an error; all of the remaining characters in the aforementioned range shall be deleted and insert mode shall be entered. If the motion command is invalid, the terminal shall be alerted, the cursor shall not be moved, and no text shall be deleted. |
| 34414 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34415 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34416 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34417 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34418 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34419 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34420 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34421 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34422 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34423 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34424 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34425 | <b>C</b>              | Delete from the current character to the end of the line and enter insert mode at the new end-of-line.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 34426 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34427 | <b>S</b>              | Clear the entire current command line and enter insert mode.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34428 | <b>[count]rc</b>      | Replace the current character with the character 'c'. With a number <i>count</i> , replace the current and the following <i>count</i> –1 characters. After this command, the current cursor position shall be on the last character that was changed. If the <i>count</i> is larger than the number of characters after the cursor, this shall not be considered an error; all of the remaining characters shall be changed.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34429 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34430 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34431 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 34432 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |

|       |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-------|-----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 34433 | <b>[count]_</b>       | Append a <space> character after the current character position and then append the last bigword in the previous input line after the <space> character. Then enter insert mode after the last character just appended. With a number <i>count</i> , append the <i>count</i> th bigword in the previous line.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 34434 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34435 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34436 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34437 | <b>[count]x</b>       | Delete the character at the current cursor position and place the deleted characters in the save buffer. If the cursor was positioned on the last character of the line, the character shall be deleted and the cursor position shall be moved to the previous character (the new last character). If the <i>count</i> is larger than the number of characters after the cursor, this shall not be considered an error; all the characters from the cursor to the end of the line shall be deleted.                                                                                                                                                                                                                                                                                                                                                                                         |
| 34438 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34439 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34440 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34441 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34442 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34443 | <b>[count]X</b>       | Delete the character before the current cursor position and place the deleted characters in the save buffer. The character under the current cursor position shall not change. If the cursor was positioned on the first character of the line, the terminal shall be alerted, and the <b>X</b> command shall have no effect. If the line contained a single character, the <b>X</b> command shall have no effect. If the line contained no characters, the terminal shall be alerted and the cursor shall not be moved. If the <i>count</i> is larger than the number of characters before the cursor, this shall not be considered an error; all the characters from before the cursor to the beginning of the line shall be deleted.                                                                                                                                                     |
| 34444 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34445 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34446 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34447 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34448 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34449 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34450 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34451 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34452 | <b>[count]dmotion</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34453 |                       | Delete the characters between the current cursor position and the character position that would result from the <i>motion</i> command. A number <i>count</i> repeats the <i>motion</i> command <i>count</i> times. If the <i>motion</i> command would move toward the beginning of the command line, the character under the current cursor position shall not be deleted. If the <i>motion</i> command is <b>d</b> , the entire current command line shall be cleared. If the <i>count</i> is larger than the number of characters between the current cursor position and the end of the command line toward which the motion command would move the cursor, this shall not be considered an error; all of the remaining characters in the aforementioned range shall be deleted. The deleted characters shall be placed in the save buffer.                                              |
| 34454 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34455 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34456 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34457 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34458 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34459 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34460 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34461 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34462 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34463 | <b>D</b>              | Delete all characters from the current cursor position to the end of the line. The deleted characters shall be placed in the save buffer.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 34464 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34465 | <b>[count]ymotion</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34466 |                       | Yank (that is, copy) the characters from the current cursor position to the position resulting from the <i>motion</i> command into a save buffer. A number <i>count</i> shall be applied to the <i>motion</i> command. If the <i>motion</i> command would move toward the beginning of the command line, the character under the current cursor position shall not be included in the set of yanked characters. If the <i>motion</i> command is <b>y</b> , the entire current command line shall be yanked into the save buffer. The current cursor position shall be unchanged. If the <i>count</i> is larger than the number of characters between the current cursor position and the end of the command line toward which the motion command would move the cursor, this shall not be considered an error; all of the remaining characters in the aforementioned range shall be yanked. |
| 34467 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34468 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34469 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34470 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34471 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34472 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34473 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34474 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34475 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34476 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34477 | <b>Y</b>              | Yank the characters from the current cursor position to the end of the line into the save buffer. The current character position shall be unchanged.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 34478 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34479 | <b>[count]p</b>       | Put a copy of the current contents of the save buffer after the current cursor position. The current cursor position shall be advanced to the last character put from the save buffer. A <i>count</i> shall indicate how many copies of the save buffer                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 34480 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 34481 |                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |

|       |                               |                                                                                                                                                                                                                                                                |
|-------|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 34482 |                               | shall be put.                                                                                                                                                                                                                                                  |
| 34483 | <b>[count]P</b>               | Put a copy of the current contents of the save buffer before the current cursor position. The current cursor position shall be moved to the last character put from the save buffer. A <i>count</i> shall indicate how many copies of the save buffer shall be |
| 34484 |                               | put.                                                                                                                                                                                                                                                           |
| 34485 |                               |                                                                                                                                                                                                                                                                |
| 34486 |                               |                                                                                                                                                                                                                                                                |
| 34487 | <b>u</b>                      | Undo the last command that modified the text of the current command line.                                                                                                                                                                                      |
| 34488 | <b>U</b>                      | Undo all changes made to the current command line since first entering command                                                                                                                                                                                 |
| 34489 |                               | mode on the line.                                                                                                                                                                                                                                              |
| 34490 | <b>[count]k</b>               |                                                                                                                                                                                                                                                                |
| 34491 | <b>[count]–</b>               | Replace the current command line with the previous command line in the shell                                                                                                                                                                                   |
| 34492 |                               | command history. The cursor shall be positioned on the first character of the new                                                                                                                                                                              |
| 34493 |                               | command. A count preceding the command shall have the same effect as                                                                                                                                                                                           |
| 34494 |                               | executing the command <i>count</i> times. If a <b>k</b> or <b>–</b> command retreats past the                                                                                                                                                                  |
| 34495 |                               | maximum number of commands in effect for this shell (affected by the <i>HISTSIZE</i>                                                                                                                                                                           |
| 34496 |                               | environment variable), the terminal shall be alerted and the command shall have                                                                                                                                                                                |
| 34497 |                               | no effect.                                                                                                                                                                                                                                                     |
| 34498 | <b>[count]j</b>               |                                                                                                                                                                                                                                                                |
| 34499 | <b>[count]+</b>               | Replace the current command line with the next command line in the shell                                                                                                                                                                                       |
| 34500 |                               | command history. The cursor shall be positioned on the first character of the new                                                                                                                                                                              |
| 34501 |                               | command. The command history position shall be remembered, and any <b>k</b> or <b>–</b>                                                                                                                                                                        |
| 34502 |                               | command, or <b>j</b> or <b>+</b> command, shall decrement or increment that position and then                                                                                                                                                                  |
| 34503 |                               | shall fetch the line at the new position. If a <b>j</b> or <b>+</b> command advances past the most                                                                                                                                                             |
| 34504 |                               | recent line in the history, the current command line shall be restored to the                                                                                                                                                                                  |
| 34505 |                               | contents before the first <b>k</b> or <b>–</b> .                                                                                                                                                                                                               |
| 34506 | <b>[number]G</b>              | Replace the current command line with the contents of the oldest command line                                                                                                                                                                                  |
| 34507 |                               | stored in the shell command history. With a number <i>number</i> , replace the current                                                                                                                                                                         |
| 34508 |                               | command line with the contents of command <i>number</i> in the history.                                                                                                                                                                                        |
| 34509 | <b>/string&lt;newline&gt;</b> |                                                                                                                                                                                                                                                                |
| 34510 |                               | Move backward through the command history, searching for the specified <i>string</i> ,                                                                                                                                                                         |
| 34511 |                               | beginning with the previous command line. If it is not found, the current                                                                                                                                                                                      |
| 34512 |                               | command line shall be unchanged. If it is found in a previous line, this command                                                                                                                                                                               |
| 34513 |                               | shall behave equivalently to a set of <b>k</b> commands to reach that line. If <i>string</i> begins                                                                                                                                                            |
| 34514 |                               | with ' <b>^</b> ', the characters after the ' <b>^</b> ' shall be matched only at the beginning of a                                                                                                                                                           |
| 34515 |                               | line.                                                                                                                                                                                                                                                          |
| 34516 | <b>?string&lt;newline&gt;</b> |                                                                                                                                                                                                                                                                |
| 34517 |                               | Move forward through the command history, searching for the specified string. If                                                                                                                                                                               |
| 34518 |                               | it is not found, the current command line shall be unchanged. If the string is found                                                                                                                                                                           |
| 34519 |                               | in the current command line, the current cursor position shall be moved to the                                                                                                                                                                                 |
| 34520 |                               | beginning of that string. If it is found in the history, this command shall behave                                                                                                                                                                             |
| 34521 |                               | equivalently to a set of <b>j</b> commands to reach that line. If <i>string</i> begins with ' <b>^</b> ', the                                                                                                                                                  |
| 34522 |                               | characters after the ' <b>^</b> ' shall be matched only at the beginning of a line.                                                                                                                                                                            |
| 34523 | <b>n</b>                      | Repeat the most recent / or ? command.                                                                                                                                                                                                                         |
| 34524 | <b>N</b>                      | Repeat the most recent / or ? command, reversing the direction of the search.                                                                                                                                                                                  |

34525 **EXIT STATUS**

34526 The following exit values shall be returned:

- 34527           0    The script to be executed consisted solely of zero or more blank lines or comments, or  
 34528                both.  
 34529        1-125    A non-interactive shell detected a syntax, redirection or variable assignment error.  
 34530        127    A specified *command\_file* could not be found by a non-interactive shell.

34531           Otherwise, the shell shall return the exit status of the last command it invoked or attempted to  
 34532           invoke (see also the *exit* utility in Section 2.14 on page 96).

34533 **CONSEQUENCES OF ERRORS**

34534 See Section 2.8.1 on page 65.

34535 **APPLICATION USAGE**

34536           Standard input and standard error are the files that determine whether a shell is interactive  
 34537           when **-i** is not specified. For example:

34538 `sh > file`

34539 and:

34540 `sh 2> file`

34541           create interactive and non-interactive shells, respectively. Although both accept terminal input,  
 34542           the results of error conditions are different, as described in Section 2.8.1 on page 65; in the second  
 34543           example a redirection error encountered by a special built-in utility aborts the shell.

34544           On systems that support set-user-ID scripts, a historical trapdoor has been to link a script to the  
 34545           name **-i**. When it is called by a sequence such as:

34546 `sh -`

34547 or by:

34548 `#! /bin/sh -`

34549           the historical systems have assumed that no option letters follow. Thus, this volume of  
 34550           IEEE Std. 1003.1-200x allows the single hyphen to mark the end of the options, in addition to the  
 34551           use of the regular `"—"` argument, because the older practice is so pervasive.

34552           A portable application must protect its first operand, if it starts with a plus sign, by preceding it  
 34553           with the `"—"` argument that denotes the end of the options.

34554 **EXAMPLES**

- 34555           1. Execute a shell command from a string:

34556 `sh -c "cat myfile"`

- 34557           2. Execute a shell script from a file in the current directory:

34558 `sh my_shell_cmds`34559 **RATIONALE**34560 The *sh* utility and the *set* special built-in utility share a common set of options.

34561           The KornShell ignores the contents of *IFS* upon entry to the script. A conforming application  
 34562           cannot rely on importing *IFS*. One justification for this, beyond security considerations, is to  
 34563           assist possible future shell compilers. Allowing *IFS* to be imported from the environment  
 34564           prevents many optimizations that might otherwise be performed via dataflow analysis of the  
 34565           script itself.



34566 The text in the STDIN section about non-blocking reads concerns an instance of *sh* that has been  
34567 invoked, probably by a C-language program, with standard input that has been opened using  
34568 the `O_NONBLOCK` flag; see *open()* in the System Interfaces volume of IEEE Std. 1003.1-200x. If  
34569 the shell did not reset this flag, it would immediately terminate because no input data would be  
34570 available yet and that would be considered the same as end-of-file.

34571 The options associated with a *restricted shell* (command name *rsh* and the `-r` option) were  
34572 excluded because the standard developers considered that the implied level of security could  
34573 not be achieved and they did not want to raise false expectations.

34574 On systems that support set-user-ID scripts, a historical trapdoor has been to link a script to the  
34575 name `-i`. When it is called by a sequence such as *sh -* or by `#!/bin/sh -`, the historical systems  
34576 have assumed that no option letters follow. Thus, this volume of IEEE Std. 1003.1-200x allows  
34577 the single hyphen to mark the end of the options, in addition to the use of the regular `"—"`  
34578 argument, because it was considered that the older practice was so pervasive. An alternative  
34579 approach is taken by the KornShell, where real and effective user/group IDs must match for an  
34580 interactive shell; this behavior is specifically allowed by this volume of IEEE Std. 1003.1-200x.

34581 **Note:** There are other problems with set-user-ID scripts that the two approaches described  
34582 here do not resolve.

34583 The default messages for the various *MAIL*-related messages are unspecified because they vary  
34584 across implementations. Typical messages are:

34585 "you have mail\n"

34586 or:

34587 "you have new mail\n"

34588 It is important that the descriptions of command line editing refer to the same shell as that in the  
34589 base standard so that interactive users can also be application programmers without having to  
34590 deal with programmatic differences in their two environments. It is also essential that the utility  
34591 name *sh* be specified because this explicit utility name is too firmly rooted in historical practice  
34592 of application programs for it to change.

34593 Consideration was given to mandating a diagnostic message when attempting to set *vi*-mode on  
34594 terminals that do not support command line editing. However, it is not historical practice for the  
34595 shell to be cognizant of all terminal types and thus be able to detect inappropriate terminals in  
34596 all cases. Implementations are encouraged to supply diagnostics in this case whenever possible,  
34597 rather than leaving the user in a state where editing commands work incorrectly.

34598 In early proposals, the KornShell-derived *emacs* mode of command line editing was included,  
34599 even though the *emacs* editor itself was not. The community of *emacs* proponents was adamant  
34600 that the full *emacs* editor not be included in this volume of IEEE Std. 1003.1-200x because they  
34601 were concerned that an attempt to standardize this very powerful environment would  
34602 encourage vendors to ship versions conforming strictly to this volume of IEEE Std. 1003.1-200x,  
34603 but lacking the extensibility required by the community. The author of the original *emacs*  
34604 program also expressed his desire to omit the program. Furthermore, there were a number of  
34605 historical systems that did not include *emacs*, or included it without supporting it, but there were  
34606 very few that did not include and support *vi*. The shell *emacs* command line editing mode was  
34607 finally omitted from this volume of IEEE Std. 1003.1-200x because it became apparent that the  
34608 KornShell version and the editor being distributed with the GNU system had diverged in some  
34609 respects. The author of *emacs* requested that the POSIX *emacs* mode either be deleted or have a  
34610 significant number of unspecified conditions. Although the KornShell author agreed to consider  
34611 changes to bring the shell into alignment, the standard developers decided to defer specification  
34612 at this time, rather than attempting to agree on a specific subset of *emacs* late within the

34613 development of this volume of IEEE Std. 1003.1-200x. It is assumed that the *emacs* and KornShell  
 34614 developers will converge on a definition acceptable to both groups, and this may be used as a  
 34615 model for a future version of this volume of IEEE Std. 1003.1-200x. In the interim,  
 34616 implementations are free to offer additional command line editing modes based on the exact  
 34617 models of editors their users are most comfortable with.

34618 Early proposals had the following list entry in **vi Line Editing Insert Mode** on page 893:

34619 \ If followed by the *erase* or *kill* character, that character shall be inserted into the input line.  
 34620 Otherwise, the backslash itself shall be inserted into the input line.

34621 However, this is not actually a feature of *sh* command line editing insert mode, but one of some  
 34622 historical terminal line drivers. Some conforming implementations continue to do this when the  
 34623 *stty ixtextn* flag is set.

#### 34624 FUTURE DIRECTIONS

34625 None.

#### 34626 SEE ALSO

34627 *cd*, *echo*, *pwd*, *test*, *umask*, the System Interfaces volume of IEEE Std. 1003.1-200x, *dup()*, *exec*,  
 34628 *exit()*, *fork()*, *pipe()*, *signal()*, *system()*, *ulimit()*, *umask()*, *wait()*

#### 34629 CHANGE HISTORY

34630 First released in Issue 2.

#### 34631 Issue 4

34632 Aligned with the ISO/IEC 9945-2: 1993 standard.

34633 Description of the shell command language and special built-ins moved to Chapter 2 on page 35.

#### 34634 Issue 5

34635 FUTURE DIRECTIONS section added.

34636 Text is added to the DESCRIPTION for the Large File Summit proposal.

#### 34637 Issue 6

34638 The Open Group corrigenda item U029/2 has been applied, correcting the second SYNOPSIS.

34639 The Open Group corrigenda item U027/3 has been applied, correcting a typographical error.

34640 The following new requirements on POSIX implementations derive from alignment with the  
 34641 Single UNIX Specification:

34642 • The option letters derived from the set special built-in are also accepted with a leading plus  
 34643 sign (' + ').

34644 • Large file extensions are added:

34645 — Path name expansion does not fail due to the size of a file.

34646 — Shell input and output redirections have an implementation-dependent offset maximum  
 34647 that is established in the open file description.

34648 In the ENVIRONMENT VARIABLES section, the text “user’s home directory” is updated to  
 34649 “directory referred to by the *HOME* environment variable”.

34650 Descriptions for the *ENV* and *PWD* environment variables are included to align with the  
 34651 IEEE P1003.2b draft standard.

34652 The normative text is reworded to avoid use of the term “must” for application requirements.

34653 **NAME**

34654           sleep — suspend execution for an interval

34655 **SYNOPSIS**34656           sleep *time*34657 **DESCRIPTION**34658           The *sleep* utility shall suspend execution for at least the integral number of seconds specified by  
34659           the *time* operand.34660 **OPTIONS**

34661           None.

34662 **OPERANDS**

34663           The following operand shall be supported:

34664           *time*           A non-negative decimal integer specifying the number of seconds for which to  
34665                               suspend execution.34666 **STDIN**

34667           Not used.

34668 **INPUT FILES**

34669           None.

34670 **ENVIRONMENT VARIABLES**34671           The following environment variables shall affect the execution of *sleep*:34672           *LANG*           Provide a default value for the internationalization variables that are unset or null.  
34673                               If *LANG* is unset or null, the corresponding value from the implementation-  
34674                               dependent default locale shall be used. If any of the internationalization variables  
34675                               contains an invalid setting, the utility shall behave as if none of the variables had  
34676                               been defined.34677           *LC\_ALL*       If set to a non-empty string value, override the values of all the other  
34678                               internationalization variables.34679           *LC\_CTYPE*   Determine the locale for the interpretation of sequences of bytes of text data as  
34680                               characters (for example, single-byte as opposed to multi-byte characters in  
34681                               arguments).34682           *LC\_MESSAGES*34683                               Determine the locale that should be used to affect the format and contents of  
34684                               diagnostic messages written to standard error.34685 *XSI*           *NLSPATH*   Determine the location of message catalogs for the processing of *LC\_MESSAGES*.34686 **ASYNCHRONOUS EVENTS**34687           If the *sleep* utility receives a SIGALRM signal, one of the following actions shall be taken:

- 34688           1. Terminate normally with a zero exit status.
- 
- 34689           2. Effectively ignore the signal.
- 
- 34690           3. Provide the default behavior for signals described in the ASYNCHRONOUS EVENTS
- 
- 34691                               section of Section 1.11 on page 25. This could include terminating with a non-zero exit
- 
- 34692                               status.

34693           The *sleep* utility shall take the standard action for all other signals.

34694 **STDOUT**

34695 Not used.

34696 **STDERR**

34697 Used only for diagnostic messages.

34698 **OUTPUT FILES**

34699 None.

34700 **EXTENDED DESCRIPTION**

34701 None.

34702 **EXIT STATUS**

34703 The following exit values shall be returned:

34704       0 The execution was successfully suspended for at least *time* seconds, or a SIGALRM signal  
34705       was received. See the ASYNCHRONOUS EVENTS section.

34706       &gt;0 An error occurred.

34707 **CONSEQUENCES OF ERRORS**

34708 Default.

34709 **APPLICATION USAGE**

34710 None.

34711 **EXAMPLES**34712 The *sleep* utility can be used to execute a command after a certain amount of time, as in:34713 (sleep 105; *command*) &

34714 or to execute a command every so often, as in:

34715 while true

34716 do

34717       *command*

34718       sleep 37

34719 done

34720 **RATIONALE**

34721 The exit status is allowed to be zero when *sleep* is interrupted by the SIGALRM signal because  
34722 most implementations of this utility rely on the arrival of that signal to notify them that the  
34723 requested finishing time has been successfully attained. Such implementations thus do not  
34724 distinguish this situation from the successful completion case. Other implementations are  
34725 allowed to catch the signal and go back to sleep until the requested time expires or to provide  
34726 the normal signal termination procedures.

34727 As with all other utilities that take integral operands and do not specify subranges of allowed  
34728 values, *sleep* is required by this volume of IEEE Std. 1003.1-200x to deal with *time* requests of up  
34729 to 2 147 483 647 seconds. This may mean that some implementations have to make multiple calls  
34730 to the delay mechanism of the underlying operating system if its argument range is less than  
34731 this.

34732 **FUTURE DIRECTIONS**

34733 None.

34734 **SEE ALSO**34735 *wait*, the System Interfaces volume of IEEE Std. 1003.1-200x, *alarm()*, *sleep()*

34736 **CHANGE HISTORY**

34737 First released in Issue 2.

34738 **Issue 4**

34739 Aligned with the ISO/IEC 9945-2: 1993 standard.

## 34740 NAME

34741        `sort` — sort, merge, or sequence check text files

## 34742 SYNOPSIS

34743        `sort [-m][-o output][-bdfinru][-t char][-k keydef]... [file...]`

34744        `sort -c [-bdfinru][-t char][-k keydef]... [file...]`

## 34745 DESCRIPTION

34746        The *sort* utility shall perform one of the following functions:

- 34747        1. Sort lines of all the named files together and write the result to the specified output.
- 34748        2. Merge lines of all the named (presorted) files together and write the result to the specified  
34749        output.
- 34750        3. Check that a single input file is correctly presorted.

34751        Comparisons shall be based on one or more sort keys extracted from each line of input (or the  
34752        entire line if no sort keys are specified), and shall be performed using the collating sequence of  
34753        the current locale.

## 34754 OPTIONS

34755        The *sort* utility shall conform to the System Interface Definitions volume of  
34756        IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines, and the `-k keydef` option should  
34757        follow the `-b`, `-d`, `-f`, `-i`, `-n`, and `-r` options.

34758        The following options shall be supported:

- |                         |                         |                                                                                                                                                                                                                         |
|-------------------------|-------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 34759<br>34760<br>34761 | <b>-c</b>               | Check that the single input file is ordered as specified by the arguments and the collating sequence of the current locale. No output shall be produced; only the exit code shall be affected.                          |
| 34762                   | <b>-m</b>               | Merge only; the input file shall be assumed to be already sorted.                                                                                                                                                       |
| 34763<br>34764          | <b>-o <i>output</i></b> | Specify the name of an output file to be used instead of the standard output. This file can be the same as one of the input <i>files</i> .                                                                              |
| 34765<br>34766<br>34767 | <b>-u</b>               | Unique: suppress all but one in each set of lines having equal keys. If used with the <code>-c</code> option, check that there are no lines with duplicate keys, in addition to checking that the input file is sorted. |

34768        The following options shall override the default ordering rules. When ordering options appear  
34769        independent of any key field specifications, the requested field ordering rules shall be applied  
34770        globally to all sort keys. When attached to a specific key (see `-k`), the specified ordering options  
34771        shall override all global ordering options for that key.

- |                         |           |                                                                                                                                                                                                                                                                 |
|-------------------------|-----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 34772<br>34773<br>34774 | <b>-d</b> | Specify that only <blank> characters and alphanumeric characters, according to the current setting of <i>LC_CTYPE</i> , shall be significant in comparisons. The behavior is undefined for a sort key to which <code>-i</code> or <code>-n</code> also applies. |
| 34775<br>34776<br>34777 | <b>-f</b> | Consider all lowercase characters that have uppercase equivalents, according to the current setting of <i>LC_CTYPE</i> , to be the uppercase equivalent for the purposes of comparison.                                                                         |
| 34778<br>34779          | <b>-i</b> | Ignore all characters that are non-printable, according to the current setting of <i>LC_CTYPE</i> .                                                                                                                                                             |
| 34780<br>34781<br>34782 | <b>-n</b> | Restrict the sort key to an initial numeric string, consisting of optional <blank> characters, optional minus sign, and zero or more digits with an optional radix character and thousands separators (as defined in the current locale), which shall           |

34783 be sorted by arithmetic value. An empty digit string shall be treated as zero.  
 34784 Leading zeros and signs on zeros shall not affect ordering.

34785 **-r** Reverse the sense of comparisons.

34786 The treatment of field separators can be altered using the options:

34787 **-b** Ignore leading <blank> characters when determining the starting and ending  
 34788 positions of a restricted sort key. If the **-b** option is specified before the first **-k**  
 34789 option, it shall be applied to all **-k** options. Otherwise, the **-b** option can be  
 34790 attached independently to each **-k** *field\_start* or *field\_end* option-argument (see  
 34791 below).

34792 **-t char** Use *char* as the field separator character; *char* shall not be considered to be part of a  
 34793 field (although it can be included in a sort key). Each occurrence of *char* shall be  
 34794 significant (for example, <*char*><*char*> delimits an empty field). If **-t** is not  
 34795 specified, <blank> characters shall be used as default field separators; each  
 34796 maximal non-empty sequence of <blank> characters that follows a non-<blank>  
 34797 character shall be a field separator.

34798 Sort keys can be specified using the options:

34799 **-k keydef** The *keydef* argument is a restricted sort key field definition. The format of this  
 34800 definition is:

34801 *field\_start*[*type*][,*field\_end*[*type*]]

34802 where *field\_start* and *field\_end* define a key field restricted to a portion of the line  
 34803 (see the EXTENDED DESCRIPTION section), and *type* is a modifier from the list of  
 34804 characters 'b', 'd', 'f', 'i', 'n', 'r'. The 'b' modifier shall behave like the  
 34805 **-b** option, but applies only to the *field\_start* or *field\_end* to which it is attached. The  
 34806 other modifiers shall behave like the corresponding options, but shall apply only  
 34807 to the key field to which they are attached; they shall have this effect if specified  
 34808 with *field\_start*, *field\_end*, or both. If any modifier is attached to a *field\_start* or to a  
 34809 *field\_end*, no option shall apply to either. Implementations shall support at least  
 34810 nine occurrences of the **-k** option, which shall be significant in command line  
 34811 order. If no **-k** option is specified, a default sort key of the entire line shall be used.

34812 When there are multiple key fields, later keys shall be compared only after all  
 34813 earlier keys compare equal. Except when the **-u** option is specified, lines that  
 34814 otherwise compare equal shall be ordered as if none of the options **-d**, **-f**, **-i**, **-n**, or  
 34815 **-k** were present (but with **-r** still in effect, if it was specified) and with all bytes in  
 34816 the lines significant to the comparison. The order in which lines that still compare  
 34817 equal are written is unspecified.

34818 **OPERANDS**

34819 The following operand shall be supported:

34820 *file* A path name of a file to be sorted, merged, or checked. If no *file* operands are  
 34821 specified, or if a *file* operand is '-', the standard input shall be used.

34822 **STDIN**

34823 The standard input shall be used only if no *file* operands are specified, or if a *file* operand is '-'.  
 34824 See the INPUT FILES section.

34825 **INPUT FILES**

34826 The input files shall be text files, except that the *sort* utility shall add a <newline> character to  
 34827 the end of a file ending with an incomplete last line.

34828 **ENVIRONMENT VARIABLES**

34829 The following environment variables shall affect the execution of *sort*:

34830 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 34831 If *LANG* is unset or null, the corresponding value from the implementation-  
 34832 dependent default locale shall be used. If any of the internationalization variables  
 34833 contains an invalid setting, the utility shall behave as if none of the variables had  
 34834 been defined.

34835 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 34836 internationalization variables.

34837 *LC\_COLLATE*  
 34838 Determine the locale for ordering rules.

34839 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 34840 characters (for example, single-byte as opposed to multi-byte characters in  
 34841 arguments and input files) and the behavior of character classification for the *-b*,  
 34842 *-d*, *-f*, *-i*, and *-n* options.

34843 *LC\_MESSAGES*  
 34844 Determine the locale that should be used to affect the format and contents of  
 34845 diagnostic messages written to standard error.

34846 *LC\_NUMERIC*  
 34847 Determine the locale for the definition of the radix character and thousands  
 34848 separator for the *-n* option.

34849 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

34850 **ASYNCHRONOUS EVENTS**

34851 Default.

34852 **STDOUT**

34853 Unless the *-o* or *-c* options are in effect, the standard output shall contain the sorted input.

34854 **STDERR**

34855 Used for diagnostic messages. A warning message about correcting an incomplete last line of an  
 34856 input file may be generated, but need not affect the final exit status.

34857 **OUTPUT FILES**

34858 If the *-o* option is in effect, the sorted input shall be placed in the file *output*.

34859 **EXTENDED DESCRIPTION**

34860 The notation:

34861 *-k field\_start[type][,field\_end[type]]*

34862 shall define a key field that begins at *field\_start* and ends at *field\_end* inclusive, unless *field\_start*  
 34863 falls beyond the end of the line or after *field\_end*, in which case the key field is empty. A missing  
 34864 *field\_end* shall mean the last character of the line.

34865 A field comprises a maximal sequence of non-separating characters and, in the absence of option  
 34866 *-t*, any preceding field separator.

34867 The *field\_start* portion of the *keydef* option-argument shall have the form:



34868 *field\_number*[*.first\_character*]

34869 Fields and characters within fields shall be numbered starting with 1. The *field\_number* and  
 34870 *first\_character* pieces, interpreted as positive decimal integers, shall specify the first character to  
 34871 be used as part of a sort key. If *.first\_character* is omitted, it shall refer to the first character of the  
 34872 field.

34873 The *field\_end* portion of the *keydef* option-argument shall have the form:

34874 *field\_number*[*.last\_character*]

34875 The *field\_number* shall be as described above for *field\_start*. The *last\_character* piece, interpreted  
 34876 as a non-negative decimal integer, shall specify the last character to be used as part of the sort  
 34877 key. If *last\_character* evaluates to zero or *.last\_character* is omitted, it shall refer to the last  
 34878 character of the field specified by *field\_number*.

34879 If the **-b** option or **b** type modifier is in effect, characters within a field shall be counted from the  
 34880 first non-<blank> character in the field. (This shall apply separately to *first\_character* and  
 34881 *last\_character*.)

## 34882 EXIT STATUS

34883 The following exit values shall be returned:

34884     0 All input files were output successfully, or **-c** was specified and the input file was correctly  
 34885 sorted.

34886     1 Under the **-c** option, the file was not ordered as specified, or if the **-c** and **-u** options were  
 34887 both specified, two input lines were found with equal keys.

34888     >1 An error occurred.

## 34889 CONSEQUENCES OF ERRORS

34890 Default.

## 34891 APPLICATION USAGE

34892 The default value for **-t**, <blank> character, has different properties from, for example, **-t**  
 34893 "<space>". If a line contains:

34894 <space><space>foo

34895 the following treatment would occur with default separation as opposed to specifically selecting  
 34896 a <space> character:

| Field | Default           | <b>-t</b> "<space>" |
|-------|-------------------|---------------------|
| 1     | <space><space>foo | <i>empty</i>        |
| 2     | <i>empty</i>      | <i>empty</i>        |
| 3     | <i>empty</i>      | foo                 |

34901 The leading field separator itself is included in a field when **-t** is not used. For example, this  
 34902 command returns an exit status of zero, meaning the input was already sorted:

34903 sort -c -k 2 <<eof

34904 y<tab>b

34905 x<space>a

34906 eof

34907 (assuming that a <tab> character precedes the <space> character in the current collating  
 34908 sequence). The field separator is not included in a field when it is explicitly set via **-t**. This is  
 34909 historical practice and allows usage such as:

```

34910 sort -t "|" -k 2n <<eof
34911 Atlanta|425022|Georgia
34912 Birmingham|284413|Alabama
34913 Columbia|100385|South Carolina
34914 eof

```

34915 where the second field can be correctly sorted numerically without regard to the non-numeric  
 34916 field separator.

34917 The wording in the OPTIONS section clarifies that the **-b**, **-d**, **-f**, **-i**, **-n**, and **-r** options have to  
 34918 come before the first sort key specified if they are intended to apply to all specified keys. The  
 34919 way it is described in this volume of IEEE Std. 1003.1-200x matches historical practice, not  
 34920 historical documentation. In the non-obsolescent versions, the results are unspecified if these  
 34921 options are specified after a **-k** option.

34922 The **-f** option might not work as expected in locales where there is not a one-to-one mapping  
 34923 between an uppercase and a lowercase letter.

#### 34924 EXAMPLES

- 34925 1. The following command sorts the contents of **infile** with the second field as the sort key:  
 34926 

```
sort -k 2,2 infile
```
- 34927 2. The following command sorts, in reverse order, the contents of **infile1** and **infile2**, placing  
 34928 the output in **outfile** and using the second character of the second field as the sort key  
 34929 (assuming that the first character of the second field is the field separator):  
 34930 

```
sort -r -o outfile -k 2.2,2.2 infile1 infile2
```
- 34931 3. The following command sorts the contents of **infile1** and **infile2** using the second non-  
 34932 <blank> character of the second field as the sort key:  
 34933 

```
sort -k 2.2b,2.2b infile1 infile2
```
- 34934 4. The following command prints the System V password file (user database) sorted by the  
 34935 numeric user ID (the third colon-separated field):  
 34936 

```
sort -t : -k 3,3n /etc/passwd
```
- 34937 5. The following command prints the lines of the already sorted file **infile**, suppressing all  
 34938 but one occurrence of lines having the same third field:  
 34939 

```
sort -um -k 3.1,3.0 infile
```

#### 34940 RATIONALE

34941 Examples in some historical documentation state that options **-um** with one input file keep the  
 34942 first in each set of lines with equal keys. This behavior was deemed to be an implementation  
 34943 artifact and was not standardized.

34944 The **-z** option was omitted; it is not standard practice on most systems and is inconsistent with  
 34945 using *sort* to sort several files individually and then merge them together. The text concerning **-z**  
 34946 in historical documentation appeared to require implementations to determine the proper buffer  
 34947 length during the sort phase of operation, but not during the merge.

34948 The **-y** option was omitted because of non-portability. The **-M** option, present in System V, was  
 34949 omitted because of non-portability in international usage.

34950 An undocumented **-T** option exists in some implementations. It is used to specify a directory for  
 34951 intermediate files. Implementations are encouraged to support the use of the *TMPCDIR*  
 34952 environment variable instead of adding an option to support this functionality.

34953 The **-k** option was added to satisfy two objections. First, the zero-based counting used by *sort* is  
34954 not consistent with other utility conventions. Second, it did not meet syntax guideline  
34955 requirements.

34956 Historical documentation indicates that “setting **-n** implies **-b**”. The description of **-n** already  
34957 states that optional leading <blank>s are tolerated in doing the comparison. If **-b** is enabled,  
34958 rather than implied, by **-n**, this has unusual side effects. When a character offset is used in a  
34959 column of numbers (for example, to sort modulo 100), that offset is measured relative to the  
34960 most significant digit, not to the column. Based upon a recommendation from the author of the  
34961 original *sort* utility, the **-b** implication has been omitted from this volume of  
34962 IEEE Std. 1003.1-200x, and an application wishing to achieve the previously mentioned side  
34963 effects has to code the **-b** flag manually.

#### 34964 **FUTURE DIRECTIONS**

34965 None.

#### 34966 **SEE ALSO**

34967 *comm*, *join*, *uniq*, the System Interfaces volume of IEEE Std. 1003.1-200x, *toupper()*

#### 34968 **CHANGE HISTORY**

34969 First released in Issue 2.

#### 34970 **Issue 4**

34971 Aligned with the ISO/IEC 9945-2:1993 standard.

## 34972 NAME

34973 split — split files into pieces

## 34974 SYNOPSIS

34975 UP `split [-l line_count][-a suffix_length][file[name]]`34976 `split -b n[k|m][-a suffix_length][file[name]]`

34977

## 34978 DESCRIPTION

34979 The *split* utility shall read an input file and write one or more output files. The default size of  
 34980 each output file shall be 1 000 lines. The size of the output files can be modified by specification  
 34981 of the **-b** or **-l** options. Each output file shall be created with a unique suffix. The suffix shall  
 34982 consist of exactly *suffix\_length* lowercase letters from the POSIX locale. The letters of the suffix  
 34983 shall be used as if they were a base-26 digit system, with the first suffix to be created consisting  
 34984 of all 'a' characters, the second with a 'b' replacing the last 'a', and so on, until a name of all  
 34985 'z' characters is created. By default, the names of the output files shall be 'x', followed by a  
 34986 two-character suffix from the character set as described above, starting with "aa", "ab", "ac",  
 34987 and so on, and continuing until the suffix "zz", for a maximum of 676 files.

34988 If the number of files required exceeds the maximum allowed by the suffix length provided,  
 34989 such that the last allowable file would be larger than the requested size, the *split* utility shall fail  
 34990 after creating the last file with a valid suffix; *split* shall not delete the files it created with valid  
 34991 suffixes. If the file limit is not exceeded, the last file created shall contain the remainder of the  
 34992 input file, and may be smaller than the requested size.

## 34993 OPTIONS

34994 The *split* utility shall conform to the System Interface Definitions volume of  
 34995 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

34996 The following options shall be supported:

34997 **-a *suffix\_length***

34998 Use *suffix\_length* letters to form the suffix portion of the file names of the split file.  
 34999 If **-a** is not specified, the default suffix length shall be two. If the sum of the *name*  
 35000 operand and the *suffix\_length* option-argument would create a file name exceeding  
 35001 {NAME\_MAX} bytes, an error shall result; *split* shall exit with a diagnostic  
 35002 message and no files shall be created.

35003 **-b *n*** Split a file into pieces *n* bytes in size.

35004 **-b *nk*** Split a file into pieces *n*\*1024 bytes in size.

35005 **-b *nm*** Split a file into pieces *n*\*1 048 576 bytes in size.

35006 **-l *line\_count*** Specify the number of lines in each resulting file piece. The *line\_count* argument is  
 35007 an unsigned decimal integer. The default is 1 000. If the input does not end with a  
 35008 <newline> character, the partial line shall be included in the last output file.

## 35009 OPERANDS

35010 The following operands shall be supported:

35011 *file* The path name of the ordinary file to be split. If no input file is given or *file* is '-',  
 35012 the standard input shall be used.

35013 *name* The prefix to be used for each of the files resulting from the split operation. If no  
 35014 *name* argument is given, 'x' shall be used as the prefix of the output files. The  
 35015 combined length of the basename of *prefix* and *suffix\_length* cannot exceed  
 35016 {NAME\_MAX} bytes. See the OPTIONS section.

35017 **STDIN**

35018 See the INPUT FILES section.

35019 **INPUT FILES**

35020 Any file can be used as input.

35021 **ENVIRONMENT VARIABLES**35022 The following environment variables shall affect the execution of *split*:

35023 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 35024 If *LANG* is unset or null, the corresponding value from the implementation-  
 35025 dependent default locale shall be used. If any of the internationalization variables  
 35026 contains an invalid setting, the utility shall behave as if none of the variables had  
 35027 been defined.

35028 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 35029 internationalization variables.

35030 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 35031 characters (for example, single-byte as opposed to multi-byte characters in  
 35032 arguments and input files).

35033 *LC\_MESSAGES*

35034 Determine the locale that should be used to affect the format and contents of  
 35035 diagnostic messages written to standard error.

35036 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

35037 **ASYNCHRONOUS EVENTS**

35038 Default.

35039 **STDOUT**

35040 Not used.

35041 **STDERR**

35042 Used only for diagnostic messages.

35043 **OUTPUT FILES**

35044 The output files contain portions of the original input file; otherwise, unchanged.

35045 **EXTENDED DESCRIPTION**

35046 None.

35047 **EXIT STATUS**

35048 The following exit values shall be returned:

35049 0 Successful completion.

35050 &gt;0 An error occurred.

35051 **CONSEQUENCES OF ERRORS**

35052 Default.

**35053 APPLICATION USAGE**

35054 Application writers should note that this utility need not be provided on systems that do not  
35055 support the User Portability Utilities option.

**35056 EXAMPLES**

35057 In the following examples **foo** is a text file that contains 5 000 lines.

35058 1. Create five files, **xaa**, **xab**, **xac**, **xad**, and **xae**:

35059 `split foo`

35060 2. Create five files, but the suffixed portion of the created files consists of three letters, **xaaa**,  
35061 **xaab**, **xaac**, **xaad**, and **xaae**:

35062 `split -a 3 foo`

35063 3. Create three files with four-letter suffixes and a supplied prefix, **bar\_aaaa**, **bar\_aaab**, and  
35064 **bar\_aaac**:

35065 `split -a 4 -l 2000 foo bar_`

35066 4. Create as many files as are necessary to contain at most 20\*1 024 bytes, each with the  
35067 default prefix of **x** and a five-letter suffix:

35068 `split -a 5 -b 20k foo`

**35069 RATIONALE**

35070 The **-b** option was added to provide a mechanism for splitting files other than by lines. While  
35071 most uses of the **-b** option are for transmitting files over networks, some believed it would have  
35072 additional uses.

35073 The **-a** option was added to overcome the limitation of being able to create only 676 files.

35074 Consideration was given to deleting this utility, using the rationale that the function provided  
35075 by this utility is available via the *csplit* utility (see *csplit* on page 314). Upon reconsideration of  
35076 the purpose of the User Portability Extension, it was decided to retain both this utility and the  
35077 *csplit* utility because users use both utilities and have historical expectations of their behavior.  
35078 Furthermore, the splitting on byte boundaries in *split* cannot be duplicated with the historical  
35079 *csplit*.

35080 The text “*split* shall not delete the files it created with valid suffixes” would normally be  
35081 assumed, but since the related utility, *csplit*, does delete files under some circumstances, the  
35082 historical behavior of *split* is made explicit to avoid misinterpretation.

**35083 FUTURE DIRECTIONS**

35084 None.

**35085 SEE ALSO**

35086 *csplit*

**35087 CHANGE HISTORY**

35088 First released in Issue 2.

**35089 Issue 4**

35090 Aligned with the ISO/IEC 9945-2: 1993 standard.

**35091 Issue 6**

35092 This utility is now marked as part of the User Portability Utilities option.

35093 The APPLICATION USAGE section is added.

35094

The obsolescent SYNOPSIS is removed.

## 35095 NAME

35096 strings — find printable strings in files

## 35097 SYNOPSIS

35098 UP strings [-a][-t *format*][-n *number*][*file...*]

35099

## 35100 DESCRIPTION

35101 **Notes to Reviewers**35102 *This section with side shading will not appear in the final copy. - Ed.*

35103 D1, XCU, ERN 342 requests that the domain of this utility exclude internationalized strings.

35104 The *strings* utility shall look for printable strings in regular files and shall write those strings to  
 35105 standard output. A printable string is any sequence of four (by default) or more printable  
 35106 characters terminated by a <newline> or NUL character. Additional implementation-dependent  
 35107 strings may be written; see *localedef*.

## 35108 OPTIONS

35109 The *strings* utility shall conform to the System Interface Definitions volume of  
 35110 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

35111 The following options shall be supported:

35112 **-a** Scan files in their entirety. If **-a** is not specified, it is implementation-dependent  
 35113 what portion of each file is scanned for strings.

35114 **-n *number*** Specify the minimum string length, where the *number* argument is a positive  
 35115 decimal integer. The default shall be 4.

35116 **-t *format*** Write each string preceded by its byte offset from the start of the file. The format  
 35117 shall be dependent on the single character used as the *format* option-argument:

35118     d The offset shall be written in decimal.

35119     o The offset shall be written in octal.

35120     x The offset shall be written in hexadecimal.

## 35121 OPERANDS

35122 The following operand shall be supported:

35123 ***file*** A path name of a regular file to be used as input. If no *file* operand is specified, the  
 35124 *strings* utility shall read from the standard input.

## 35125 STDIN

35126 See the INPUT FILES section.

## 35127 INPUT FILES

35128 The input files named by the utility arguments or the standard input shall be regular files of any  
 35129 format.

## 35130 ENVIRONMENT VARIABLES

35131 The following environment variables shall affect the execution of *strings*:

35132 **LANG** Provide a default value for the internationalization variables that are unset or null.  
 35133 If *LANG* is unset or null, the corresponding value from the implementation-  
 35134 dependent default locale shall be used. If any of the internationalization variables  
 35135 contains an invalid setting, the utility shall behave as if none of the variables had  
 35136 been defined.



35137 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 35138 internationalization variables.

35139 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 35140 characters (for example, single-byte as opposed to multi-byte characters in  
 35141 arguments and input files) and to identify printable strings.

35142 **LC\_MESSAGES**  
 35143 Determine the locale that should be used to affect the format and contents of  
 35144 diagnostic messages written to standard error.

35145 XSI **NLSPATH** Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

35146 **ASYNCHRONOUS EVENTS**  
 35147 Default.

35148 **STDOUT**  
 35149 Strings found shall be written to the standard output, one per line.  
 35150 When the **-t** option is not specified, the format of the output shall be:  
 35151 "%s", <string>  
 35152 With the **-t o** option, the format of the output shall be:  
 35153 "%o %s", <byte offset>, <string>  
 35154 With the **-t x** option, the format of the output shall be:  
 35155 "%x %s", <byte offset>, <string>  
 35156 With the **-t d** option, the format of the output shall be:  
 35157 "%d %s", <byte offset>, <string>

35158 **STDERR**  
 35159 Used only for diagnostic messages.

35160 **OUTPUT FILES**  
 35161 None.

35162 **EXTENDED DESCRIPTION**  
 35163 None.

35164 **EXIT STATUS**  
 35165 The following exit values shall be returned:  
 35166 0 Successful completion.  
 35167 >0 An error occurred.

35168 **CONSEQUENCES OF ERRORS**  
 35169 Default.

35170 **APPLICATION USAGE**

35171 By default the data area (as opposed to the text, “bss” or header areas) of a binary executable file  
35172 is scanned. Implementations document which areas are scanned.

35173 Some historical implementations do not require NUL or <newline> character terminators for  
35174 strings to permit those languages that do not use NUL as a string terminator to have their strings  
35175 written.

35176 Application writers should note that this utility need not be provided on systems that do not  
35177 support the User Portability Utilities option.

35178 **EXAMPLES**

35179 None.

35180 **RATIONALE**

35181 Apart from rationalizing the option syntax and slight difficulties with object and executable  
35182 binary files, *strings* is specified to match historical practice closely. The **-a** and **-n** options were  
35183 introduced to replace the non-conforming **-** and **-number** options.

35184 The **-o** option historically means different things on different implementations. Some use it to  
35185 mean “*offset* in decimal”, while others use it as “*offset* in octal”. Instead of trying to decide which  
35186 way would be least objectionable, the **-t** option was added. It was originally named **-O** to mean  
35187 “*offset*”, but was changed to **-t** to be consistent with *od*.

35188 The ISO C standard function *isprint()* is restricted to a domain of **unsigned char**. This volume of  
35189 IEEE Std. 1003.1-200x requires implementations to write strings as defined by the current locale.

35190 **FUTURE DIRECTIONS**

35191 None.

35192 **SEE ALSO**

35193 *nm*

35194 **CHANGE HISTORY**

35195 First released in Issue 4.

35196 **Issue 6**

35197 This utility is now marked as part of the User Portability Utilities option.

35198 The obsolescent SYNOPSIS is removed. |

35199 The normative text is reworded to avoid use of the term “must” for application requirements. |

35200 **NAME**35201 strip — remove unnecessary information from executable files (**DEVELOPMENT**)35202 **SYNOPSIS**

35203 SD strip file...

35204

35205 **DESCRIPTION**

35206 The *strip* utility shall remove from executable files named by the *file* operands any information  
 35207 the implementor deems unnecessary for execution of those files. The nature of that information  
 35208 is unspecified. The effect of *strip* shall be the same as the use of the *-s* option to *cc*, *c89*, or *fort77*.

35209 **OPTIONS**

35210 None.

35211 **OPERANDS**

35212 The following operand shall be supported:

35213 *file* A path name referring to an executable file.35214 **STDIN**

35215 Not used.

35216 **INPUT FILES**

35217 The input files shall be in the form of executable files successfully produced by any compiler  
 35218 defined by this volume of IEEE Std. 1003.1-200x.

35219 **ENVIRONMENT VARIABLES**35220 The following environment variables shall affect the execution of *strip*:

35221 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 35222 If *LANG* is unset or null, the corresponding value from the implementation-  
 35223 dependent default locale shall be used. If any of the internationalization variables  
 35224 contains an invalid setting, the utility shall behave as if none of the variables had  
 35225 been defined.

35226 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 35227 internationalization variables.

35228 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 35229 characters (for example, single-byte as opposed to multi-byte characters in  
 35230 arguments).

35231 *LC\_MESSAGES*

35232 Determine the locale that should be used to affect the format and contents of  
 35233 diagnostic messages written to standard error.

35234 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.35235 **ASYNCHRONOUS EVENTS**

35236 Default.

35237 **STDOUT**

35238 Not used.

35239 **STDERR**

35240 Used only for diagnostic messages.

35241 **OUTPUT FILES**

35242           The *strip* utility shall produce executable files of unspecified format.

35243 **EXTENDED DESCRIPTION**

35244           None.

35245 **EXIT STATUS**

35246           The following exit values shall be returned:

35247           0   Successful completion.

35248           >0   An error occurred.

35249 **CONSEQUENCES OF ERRORS**

35250           Default.

35251 **APPLICATION USAGE**

35252           None.

35253 **EXAMPLES**

35254           None.

35255 **RATIONALE**

35256           Historically, this utility has been used to remove the symbol table from an executable file. It was  
35257           included since it is known that the amount of symbolic information can amount to several  
35258           megabytes; the ability to remove it in a portable manner was deemed important, especially for  
35259           smaller systems.

35260           The behavior of *strip* is said to be the same as the *-s* option to a compiler. While the end result is  
35261           essentially the same, it is not required to be identical. The same effect can be achieved with  
35262           either *-s* during a compile or a *strip* on the final object file.

35263 **FUTURE DIRECTIONS**

35264           None.

35265 **SEE ALSO**

35266           *ar*, *c89*, *fort77*

35267 **CHANGE HISTORY**

35268           First released in Issue 2.

35269 **Issue 4**

35270           Aligned with the ISO/IEC 9945-2: 1993 standard.

35271 **Issue 6**

35272           This utility is now marked as part of the Software Development Utilities option.

35273 **NAME**

35274           stty — set the options for a terminal

35275 **SYNOPSIS**

35276           stty [ -a | -g ]

35277           stty *operands*35278 **DESCRIPTION**

35279       The *stty* utility shall set or report on terminal I/O characteristics for the device that is its  
 35280       standard input. Without options or operands specified, it shall report the settings of certain  
 35281       characteristics, usually those that differ from implementation-dependent defaults. Otherwise, it  
 35282       shall modify the terminal state according to the specified operands. Detailed information about  
 35283       the modes listed in the first five groups below are described in the System Interface Definitions  
 35284       volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal Interface. Operands in the  
 35285       Combination Modes group (see **Combination Modes** on page 927) are implemented using  
 35286       operands in the previous groups. Some combinations of operands are mutually-exclusive on  
 35287       some terminal types; the results of using such combinations are unspecified.

35288       Typical implementations of this utility require a communications line configured to use the  
 35289       **termios** interface defined in the System Interfaces volume of IEEE Std. 1003.1-200x. On systems  
 35290       where none of these lines are available, and on lines not currently configured to support the  
 35291       **termios** interface, some of the operands need not affect terminal characteristics.

35292 **OPTIONS**

35293       The *stty* utility shall conform to the System Interface Definitions volume of  
 35294       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

35295       The following options shall be supported:

35296       **-a**           Write to standard output all the current settings for the terminal.

35297       **-g**           Write to standard output all the current settings in an unspecified form that can be  
 35298       used as arguments to another invocation of the *stty* utility on the same system. The  
 35299       form used shall not contain any characters that would require quoting to avoid  
 35300       word expansion by the shell; see Section 2.6 on page 49.

35301 **OPERANDS**

35302       The following operands shall be supported to set the terminal characteristics.

35303       **Control Modes**

35304       **parenb** (**-parenb**) Enable (disable) parity generation and detection. This has the effect of setting  
 35305       (not setting) PARENB in the **termios** *c\_cflag* field, as defined in the System  
 35306       Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General  
 35307       Terminal Interface.

35308       **parodd** (**-parodd**) Select odd (even) parity. This shall have the effect of setting (not setting)  
 35309       PARODD in the **termios** *c\_cflag* field, as defined in the System Interface  
 35310       Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal  
 35311       Interface.

35312       **cs5 cs6 cs7 cs8** Select character size, if possible. This shall have the effect of setting CS5, CS6,  
 35313       CS7, and CS8, respectively, in the **termios** *c\_cflag* field, as defined in the  
 35314       System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11,  
 35315       General Terminal Interface.

35316       *number*       Set terminal baud rate to the number given, if possible. If the baud rate is set  
 35317       to zero, the modem control lines shall not be longer asserted. This shall have

|       |                         |                                                                                                        |
|-------|-------------------------|--------------------------------------------------------------------------------------------------------|
| 35318 |                         | the effect of setting the input and output <b>termios</b> baud rate values as defined                  |
| 35319 |                         | in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter                           |
| 35320 |                         | 11, General Terminal Interface.                                                                        |
| 35321 | <b>ispeed number</b>    | Set terminal input baud rate to the number given, if possible. If the input baud                       |
| 35322 |                         | rate is set to zero, the input baud rate shall be specified by the value of the                        |
| 35323 |                         | output baud rate. This shall have the effect of setting the input <b>termios</b> baud                  |
| 35324 |                         | rate values as defined in the System Interface Definitions volume of                                   |
| 35325 |                         | IEEE Std. 1003.1-200x, Chapter 11, General Terminal Interface.                                         |
| 35326 | <b>ospeed number</b>    | Set terminal output baud rate to the number given, if possible. If the output                          |
| 35327 |                         | baud rate is set to zero, the modem control lines shall no longer be asserted.                         |
| 35328 |                         | This shall have the effect of setting the output <b>termios</b> baud rate values as                    |
| 35329 |                         | defined in the System Interface Definitions volume of IEEE Std. 1003.1-200x,                           |
| 35330 |                         | Chapter 11, General Terminal Interface.                                                                |
| 35331 | <b>hupcl (-hupcl)</b>   | Stop asserting modem control lines (do not stop asserting modem control                                |
| 35332 |                         | lines) on last close. This shall have the effect of setting (not setting) HUPCL in                     |
| 35333 |                         | the <b>termios</b> <i>c_cflag</i> field, as defined in the System Interface Definitions volume         |
| 35334 |                         | of IEEE Std. 1003.1-200x, Chapter 11, General Terminal Interface.                                      |
| 35335 | <b>hup (-hup)</b>       | Same as <b>hupcl(-hupcl)</b> .                                                                         |
| 35336 | <b>cstopb (-cstopb)</b> | Use two (one) stop bits per character. This shall have the effect of setting (not                      |
| 35337 |                         | setting) CSTOPB in the <b>termios</b> <i>c_cflag</i> field, as defined in the System Interface         |
| 35338 |                         | Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal                              |
| 35339 |                         | Interface.                                                                                             |
| 35340 | <b>cread (-cread)</b>   | Enable (disable) the receiver. This shall have the effect of setting (not setting)                     |
| 35341 |                         | CREAD in the <b>termios</b> <i>c_cflag</i> field, as defined in the System Interface                   |
| 35342 |                         | Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal                              |
| 35343 |                         | Interface.                                                                                             |
| 35344 | <b>clocal (-clocal)</b> | Assume a line without (with) modem control. This shall have the effect of                              |
| 35345 |                         | setting (not setting) CLOCAL in the <b>termios</b> <i>c_cflag</i> field, as defined in the             |
| 35346 |                         | System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11,                              |
| 35347 |                         | General Terminal Interface.                                                                            |
| 35348 |                         | It is unspecified whether <i>stty</i> shall report an error if an attempt to set a Control Mode fails. |
| 35349 | <b>Input Modes</b>      |                                                                                                        |
| 35350 | <b>ignbrk (-ignbrk)</b> | Ignore (do not ignore) break on input. This shall have the effect of setting (not                      |
| 35351 |                         | setting) IGNBRK in the <b>termios</b> <i>c_iflag</i> field, as defined in the System Interface         |
| 35352 |                         | Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal                              |
| 35353 |                         | Interface.                                                                                             |
| 35354 | <b>brkint (-brkint)</b> | Signal (do not signal) INTR on break. This shall have the effect of setting (not                       |
| 35355 |                         | setting) BRKINT in the <b>termios</b> <i>c_iflag</i> field, as defined in the System Interface         |
| 35356 |                         | Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal                              |
| 35357 |                         | Interface.                                                                                             |
| 35358 | <b>ignpar (-ignpar)</b> | Ignore (do not ignore) bytes with parity errors. This shall have the effect of                         |
| 35359 |                         | setting (not setting) IGNPAR in the <b>termios</b> <i>c_iflag</i> field, as defined in the             |
| 35360 |                         | System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11,                              |
| 35361 |                         | General Terminal Interface.                                                                            |

|           |                                  |                                                                                                  |
|-----------|----------------------------------|--------------------------------------------------------------------------------------------------|
| 35362     | <b>parmrk</b> ( <b>-parmrk</b> ) |                                                                                                  |
| 35363     |                                  | Mark (do not mark) parity errors. This shall have the effect of setting (not                     |
| 35364     |                                  | setting) PARMRK in the <b>termios</b> <i>c_iflag</i> field, as defined in the System             |
| 35365     |                                  | Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General                       |
| 35366     |                                  | Terminal Interface.                                                                              |
| 35367     | <b>inpck</b> ( <b>-inpck</b> )   | Enable (disable) input parity checking. This shall have the effect of setting (not               |
| 35368     |                                  | setting) INPCK in the <b>termios</b> <i>c_iflag</i> field, as defined in the System Interface    |
| 35369     |                                  | Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal                        |
| 35370     |                                  | Interface.                                                                                       |
| 35371     | <b>istrip</b> ( <b>-istrip</b> ) | Strip (do not strip) input characters to seven bits. This shall have the effect of               |
| 35372     |                                  | setting (not setting) ISTRIP in the <b>termios</b> <i>c_iflag</i> field, as defined in the       |
| 35373     |                                  | System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11,                        |
| 35374     |                                  | General Terminal Interface.                                                                      |
| 35375     | <b>inlcr</b> ( <b>-inlcr</b> )   | Map (do not map) NL to CR on input. This shall have the effect of setting (not                   |
| 35376     |                                  | setting) INLCR in the <b>termios</b> <i>c_iflag</i> field, as defined in the System Interface    |
| 35377     |                                  | Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal                        |
| 35378     |                                  | Interface.                                                                                       |
| 35379     | <b>igncr</b> ( <b>-igncr</b> )   | Ignore (do not ignore) CR on input. This shall have the effect of setting (not                   |
| 35380     |                                  | setting) IGNCR in the <b>termios</b> <i>c_iflag</i> field, as defined in the System Interface    |
| 35381     |                                  | Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal                        |
| 35382     |                                  | Interface.                                                                                       |
| 35383     | <b>icrnl</b> ( <b>-icrnl</b> )   | Map (do not map) CR to NL on input. This shall have the effect of setting (not                   |
| 35384     |                                  | setting) ICRNL in the <b>termios</b> <i>c_iflag</i> field, as defined in the System Interface    |
| 35385     |                                  | Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal                        |
| 35386     |                                  | Interface.                                                                                       |
| 35387     | <b>ixon</b> ( <b>-ixon</b> )     | Enable (disable) START/STOP output control. Output from the system is                            |
| 35388     |                                  | stopped when the system receives STOP and started when the system receives                       |
| 35389     |                                  | START. This shall have the effect of setting (not setting) IXON in the <b>termios</b>            |
| 35390     |                                  | <i>c_iflag</i> field, as defined in the System Interface Definitions volume of                   |
| 35391     |                                  | IEEE Std. 1003.1-200x, Chapter 11, General Terminal Interface.                                   |
| 35392 XSI | <b>ixany</b> ( <b>-ixany</b> )   | Allow any character to restart output. This shall have the effect of setting (not                |
| 35393     |                                  | setting) IXANY in the <b>termios</b> <i>c_iflag</i> field, as defined in the System Interface    |
| 35394     |                                  | Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal                        |
| 35395     |                                  | Interface.                                                                                       |
| 35396     | <b>ixoff</b> ( <b>-ixoff</b> )   | Request that the system send (not send) STOP characters when the input                           |
| 35397     |                                  | queue is nearly full and START characters to resume data transmission. This                      |
| 35398     |                                  | shall have the effect of setting (not setting) IXOFF in the <b>termios</b> <i>c_iflag</i> field, |
| 35399     |                                  | as defined in the System Interface Definitions volume of                                         |
| 35400     |                                  | IEEE Std. 1003.1-200x, Chapter 11, General Terminal Interface.                                   |

|           |                            |                                                                                                 |
|-----------|----------------------------|-------------------------------------------------------------------------------------------------|
| 35401     | <b>Output Modes</b>        |                                                                                                 |
| 35402     | <b>opost (-opost)</b>      | Post-process output (do not post-process output; ignore all other output                        |
| 35403     |                            | modes). This shall have the effect of setting (not setting) OPOST in the                        |
| 35404     |                            | <b>termios</b> <i>c_oflag</i> field, as defined in the System Interface Definitions volume of   |
| 35405     |                            | IEEE Std. 1003.1-200x, Chapter 11, General Terminal Interface.                                  |
| 35406 XSI | <b>ocrnl (-ocrnl)</b>      | Map (do not map) CR to NL on output This shall have the effect of setting                       |
| 35407     |                            | (not setting) OCRNL in the <b>termios</b> <i>c_oflag</i> field, as defined in the System        |
| 35408     |                            | Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General                      |
| 35409     |                            | Terminal Interface.                                                                             |
| 35410     | <b>onocr (-onocr)</b>      | Do not (do) output CR at column zero. This shall have the effect of setting (not                |
| 35411     |                            | setting) ONOCR in the <b>termios</b> <i>c_oflag</i> field, as defined in the System Interface   |
| 35412     |                            | Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal                       |
| 35413     |                            | Interface.                                                                                      |
| 35414     | <b>onlret (-onlret)</b>    | The terminal newline key performs (does not perform) the CR function. This                      |
| 35415     |                            | shall have the effect of setting (not setting) ONLRET in the <b>termios</b> <i>c_oflag</i>      |
| 35416     |                            | field, as defined in the System Interface Definitions volume of                                 |
| 35417     |                            | IEEE Std. 1003.1-200x, Chapter 11, General Terminal Interface.                                  |
| 35418     | <b>ofill (-ofill)</b>      | Use fill characters (use timing) for delays. This shall have the effect of setting              |
| 35419     |                            | (not setting) OFILL in the <b>termios</b> <i>c_oflag</i> field, as defined in the System        |
| 35420     |                            | Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General                      |
| 35421     |                            | Terminal Interface.                                                                             |
| 35422     | <b>ofdel (-ofdel)</b>      | Fill characters are DELs (NULs). This shall have the effect of setting (not                     |
| 35423     |                            | setting) OFDEL in the <b>termios</b> <i>c_oflag</i> field, as defined in the System Interface   |
| 35424     |                            | Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal                       |
| 35425     |                            | Interface.                                                                                      |
| 35426     | <b>cr0 cr1 cr2 cr3</b>     | Select the style of delay for CRs. This shall have the effect of setting (not                   |
| 35427     |                            | setting) CRDLY to CR1, CR2, CR3, or CR4, respectively, in the <b>termios</b> <i>c_oflag</i>     |
| 35428     |                            | field, as defined in the System Interface Definitions volume of                                 |
| 35429     |                            | IEEE Std. 1003.1-200x, Chapter 11, General Terminal Interface.                                  |
| 35430     | <b>nl0 nl1</b>             | Select the style of delay for NL. This has the effect of setting (not setting)                  |
| 35431     |                            | NLDLY to NL0 or NL1, respectively, in the <b>termios</b> <i>c_oflag</i> field, as defined in    |
| 35432     |                            | the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11,                   |
| 35433     |                            | General Terminal Interface.                                                                     |
| 35434     | <b>tab0 tab1 tab2 tab3</b> |                                                                                                 |
| 35435     |                            | Select the style of delay for horizontal tabs. This shall have the effect of setting            |
| 35436     |                            | (not setting) TABDLY to TAB0, TAB1, TAB2, or TAB3, respectively, in the                         |
| 35437     |                            | <b>termios</b> <i>c_oflag</i> field, as defined in the System Interface Definitions volume of   |
| 35438     |                            | IEEE Std. 1003.1-200x, Chapter 11, General Terminal Interface. Note that TAB3                   |
| 35439     |                            | has the effect of expanding <tab>s to <space>s.                                                 |
| 35440     | <b>bs0 bs1</b>             | Select the style of delay for backspaces. This shall have the effect of setting                 |
| 35441     |                            | (not setting) BSDLY to BS0 or BS1, respectively, in the <b>termios</b> <i>c_oflag</i> field, as |
| 35442     |                            | defined in the System Interface Definitions volume of IEEE Std. 1003.1-200x,                    |
| 35443     |                            | Chapter 11, General Terminal Interface.                                                         |
| 35444     | <b>ff0 ff1</b>             | Select the style of delay for form-feeds. This shall have the effect of setting                 |
| 35445     |                            | (not setting) FFDLY to FF0 or FF1, respectively, in the <b>termios</b> <i>c_oflag</i> field, as |
| 35446     |                            | defined in the System Interface Definitions volume of IEEE Std. 1003.1-200x,                    |
| 35447     |                            | Chapter 11, General Terminal Interface.                                                         |



|       |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------|---------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 35448 | <b>vt0 vt1</b>            | Select the style of delay for vertical-tabs. This shall have the effect of setting (not setting) VTDLY to VT0 or VT1, respectively, in the <b>termios</b> <i>c_oflag</i> field, as defined in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal Interface.                                                                                           |
| 35449 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35450 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35451 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35452 | <b>Local Modes</b>        |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35453 | <b>isig (-isig)</b>       | Enable (disable) the checking of characters against the special control characters INTR, QUIT, and SUSP. This shall have the effect of setting (not setting) ISIG in the <b>termios</b> <i>c_lflag</i> field, as defined in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal Interface.                                                             |
| 35454 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35455 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35456 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35457 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35458 | <b>icanon (-icanon)</b>   | Enable (disable) canonical input (ERASE and KILL processing). This shall have the effect of setting (not setting) ICANON in the <b>termios</b> <i>c_lflag</i> field, as defined in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal Interface.                                                                                                      |
| 35459 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35460 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35461 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35462 | <b>ixexten (-ixexten)</b> | Enable (disable) any implementation-dependent special control characters not currently controlled by <b>icanon</b> , <b>isig</b> , <b>ixon</b> , or <b>ixoff</b> . This shall have the effect of setting (not setting) IEXTEN in the <b>termios</b> <i>c_lflag</i> field, as defined in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal Interface. |
| 35463 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35464 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35465 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35466 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35467 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35468 | <b>echo (-echo)</b>       | Echo back (do not echo back) every character typed. This shall have the effect of setting (not setting) ECHO in the <b>termios</b> <i>c_lflag</i> field, as defined in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal Interface.                                                                                                                  |
| 35469 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35470 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35471 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35472 | <b>echoe (-echoe)</b>     | The ERASE character visually erases (does not erase) the last character in the current line from the display, if possible. This shall have the effect of setting (not setting) ECHOE in the <b>termios</b> <i>c_lflag</i> field, as defined in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal Interface.                                          |
| 35473 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35474 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35475 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35476 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35477 | <b>echok (-echok)</b>     | Echo (do not echo) NL after KILL character. This shall have the effect of setting (not setting) ECHOK in the <b>termios</b> <i>c_lflag</i> field, as defined in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal Interface.                                                                                                                         |
| 35478 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35479 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35480 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35481 | <b>echonl (-echonl)</b>   | Echo (do not echo) NL, even if <b>echo</b> is disabled. This shall have the effect of setting (not setting) ECHONL in the <b>termios</b> <i>c_lflag</i> field, as defined in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal Interface.                                                                                                            |
| 35482 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35483 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35484 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35485 | <b>noflsh (-noflsh)</b>   | Disable (enable) flush after INTR, QUIT, SUSP. This shall have the effect of setting (not setting) NOFLSH in the <b>termios</b> <i>c_lflag</i> field, as defined in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal Interface.                                                                                                                     |
| 35486 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35487 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35488 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35489 | <b>tostop (-tostop)</b>   | Send SIGTTOU for background output. This shall have the effect of setting (not setting) TOSTOP in the <b>termios</b> <i>c_lflag</i> field, as defined in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal Interface.                                                                                                                                |
| 35490 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35491 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35492 |                           |                                                                                                                                                                                                                                                                                                                                                                                                   |

## Special Control Character Assignments

### <control>-character string

Set <control>-character to *string*. If <control>-character is one of the character sequences in the first column of the following table, the corresponding System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal Interface control character from the second column shall be recognized. This has the effect of setting the corresponding element of the **termios** *c\_cc* array (see the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 13, Headers, <**termios.h**>).

**Table 4-19** Control Character Names in *stty*

| Control Character | c_cc Subscript | Description     |
|-------------------|----------------|-----------------|
| <b>eof</b>        | VEOF           | EOF character   |
| <b>eol</b>        | VEOL           | EOL character   |
| <b>erase</b>      | VERASE         | ERASE character |
| <b>intr</b>       | VINTR          | INTR character  |
| <b>kill</b>       | VKILL          | KILL character  |
| <b>quit</b>       | VQUIT          | QUIT character  |
| <b>susp</b>       | VSUSP          | SUSP character  |
| <b>start</b>      | VSTART         | START character |
| <b>stop</b>       | VSTOP          | STOP character  |

If *string* is a single character, the control character shall be set to that character. If *string* is the two-character sequence "^\_" or the string *undef*, the control character shall be set to `_POSIX_VDISABLE`, if it is in effect for the device; if `_POSIX_VDISABLE` is not in effect for the device, it shall be treated as an error. In the POSIX locale, if *string* is a two-character sequence beginning with circumflex ('^'), and the second character is one of those listed in the "^c" column of the following table, the control character shall be set to the corresponding character value in the Value column of the table.

**Table 4-20** Circumflex Control Characters in *stty*

| ^c   | Value | ^c   | Value | ^c   | Value |
|------|-------|------|-------|------|-------|
| a, A | <SOH> | l, L | <FF>  | w, W | <ETB> |
| b, B | <STX> | m, M | <CR>  | x, X | <CAN> |
| c, C | <ETX> | n, N | <SO>  | y, Y | <EM>  |
| d, D | <EOT> | o, O | <SI>  | z, Z | <SUB> |
| e, E | <ENQ> | p, P | <DLE> | [    | <ESC> |
| f, F | <ACK> | q, Q | <DC1> | \    | <FS>  |
| g, G | <BEL> | r, R | <DC2> | ]    | <GS>  |
| h, H | <BS>  | s, S | <DC3> | ^    | <RS>  |
| i, I | <HT>  | t, T | <DC4> | _    | <US>  |
| j, J | <LF>  | u, U | <NAK> | ?    | <DEL> |
| k, K | <VT>  | v, V | <SYN> |      |       |

### **min** *number*

### **time** *number*

Set the value of **min** or **time** to *number*. MIN and TIME are used in non-canonical mode input processing (**icanon**).

35536 **Combination Modes**35537 *saved settings*35538 Set the current terminal characteristics to the saved settings produced by the **-g** option.35539 **evenp** or **parity**35540 Enable **parenb** and **cs7**; disable **parodd**.35541 **oddp**35542 Enable **parenb**, **cs7**, and **parodd**.35543 **-parity**, **-evenp**, or **-oddp**35544 Disable **parenb**, and set **cs8**.35545 XSI **raw** (**-raw** or **cooked**)

35546 Enable (disable) raw input and output. Raw mode shall be equivalent to setting:

```
35547 stty cs8 erase ^- kill ^- intr ^- \
35548 quit ^- eof ^- eol ^- -post -inpck
```

35549 **nl** (**-nl**)35550 Enable (disable) **icrnl**. In addition, **-nl** unsets **inlcr** and **igncr**.35551 XSI **tabs** (**-tabs** or **tab3**)

35552 Preserve tabs (expand to spaces) when printing.

35553 **ek** Reset ERASE and KILL characters back to system defaults.35554 **sane** Reset all modes to some reasonable, unspecified, values.35555 **STDIN**

35556 Although no input is read from standard input, standard input is used to get the current  
 35557 terminal I/O characteristics and to set new terminal I/O characteristics.

35558 **INPUT FILES**

35559 None.

35560 **ENVIRONMENT VARIABLES**35561 The following environment variables shall affect the execution of *stty*:

35562 **LANG** Provide a default value for the internationalization variables that are unset or null.  
 35563 If **LANG** is unset or null, the corresponding value from the implementation-  
 35564 dependent default locale shall be used. If any of the internationalization variables  
 35565 contains an invalid setting, the utility shall behave as if none of the variables had  
 35566 been defined.

35567 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 35568 internationalization variables.

35569 **LC\_CTYPE** This variable determines the locale for the interpretation of sequences of bytes of  
 35570 text data as characters (for example, single-byte as opposed to multi-byte  
 35571 characters in arguments) and which characters are in the class **print**.

35572 **LC\_MESSAGES**

35573 Determine the locale that should be used to affect the format and contents of  
 35574 diagnostic messages written to standard error.

35575 XSI **NLSPATH** Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

35576 **ASYNCHRONOUS EVENTS**

35577 Default.

35578 **STDOUT**

35579 If operands are specified, no output shall be produced.

35580 If the **-g** option is specified, *stty* shall write to standard output the current settings in a form that  
 35581 can be used as arguments to another instance of *stty* on the same system.

35582 If the **-a** option is specified, all of the information as described in the OPERANDS section shall  
 35583 be written to standard output. Unless otherwise specified, this information shall be written as  
 35584 <space>-separated tokens in an unspecified format, on one or more lines, with an unspecified  
 35585 number of tokens per line. Additional information may be written.

35586 If no options or operands are specified, an unspecified subset of the information written for the  
 35587 **-a** option shall be written.

35588 If speed information is written as part of the default output, or if the **-a** option is specified and if  
 35589 the terminal input speed and output speed are the same, the speed information shall be written  
 35590 as follows:

35591 "speed %d baud;", &lt;speed&gt;

35592 Otherwise, speeds shall be written as:

35593 "ispeed %d baud; ospeed %d baud;", &lt;ispeed&gt;, &lt;ospeed&gt;

35594 In locales other than the POSIX locale, the word **baud** may be changed to something more  
 35595 appropriate in those locales.

35596 If control characters are written as part of the default output, or if the **-a** option is specified,  
 35597 control characters shall be written as:

35598 "%s = %s;", &lt;control-character name&gt;, &lt;value&gt;

35599 where <value> is either the character, or some visual representation of the character if it is non-  
 35600 printable, or the string *undef* if the character is disabled.

35601 **STDERR**

35602 Used only for diagnostic messages.

35603 **OUTPUT FILES**

35604 None.

35605 **EXTENDED DESCRIPTION**

35606 None.

35607 **EXIT STATUS**

35608 The following exit values shall be returned:

35609 0 The terminal options were read or set successfully.

35610 &gt;0 An error occurred.

35611 **CONSEQUENCES OF ERRORS**

35612 Default.

35613 **APPLICATION USAGE**

35614 The **-g** flag is designed to facilitate the saving and restoring of terminal state from the shell level.  
 35615 For example, a program may:

```
35616 saveterm="$(stty -g)" # save terminal state
35617 stty (new settings) # set new state
35618 ... # ...
35619 stty $saveterm # restore terminal state
```

35620 Since the format is unspecified, the saved value is not portable across systems.

35621 Since the **-a** format is so loosely specified, scripts that save and restore terminal settings should  
 35622 use the **-g** option.

35623 **EXAMPLES**

35624 None.

35625 **RATIONALE**

35626 The original *stty* description was taken directly from System V and reflected the System V  
 35627 terminal driver **termio**. It has been modified to correspond to the terminal driver **termios**.

35628 Since the System Interface Definitions volume of IEEE Std. 1003.1-200x does not specify any  
 35629 output modes, they are not specified in this volume of IEEE Std. 1003.1-200x either.  
 35630 Implementations are expected to provide *stty* operands corresponding to all of the output modes  
 35631 they support.

35632 In many ways outside the scope of this volume of IEEE Std. 1003.1-200x, *stty* is primarily used to  
 35633 tailor the user interface of the terminal, such as selecting the preferred ERASE and KILL  
 35634 characters. As an application programming utility, *stty* can be used within shell scripts to alter  
 35635 the terminal settings for the duration of the script.

35636 The **termios** section states that individual disabling of control characters is possible through the  
 35637 option **\_POSIX\_VDISABLE**. If enabled, two conventions currently exist for specifying this:  
 35638 System V uses "**^-**", and BSD uses *undef*. Both are accepted by *stty* in this volume of  
 35639 IEEE Std. 1003.1-200x. The other BSD convention of using the letter '**u**' was rejected because it  
 35640 conflicts with the actual letter '**u**', which is an acceptable value for a control character.

35641 Early proposals did not specify the mapping of "**^c**" to control characters because the control  
 35642 characters were not specified in the POSIX locale character set description file requirements. The  
 35643 control character set is now specified in the System Interface Definitions volume of  
 35644 IEEE Std. 1003.1-200x, Chapter 3, Definitions so the historical mapping is specified. Note that  
 35645 although the mapping corresponds to control-character key assignments on many terminals that  
 35646 use the ISO/IEC 646:1991 standard (or ASCII) character encodings, the mapping specified here  
 35647 is to the control characters, not their keyboard encodings.

35648 Since **termios** supports separate speeds for input and output, two new options were added to  
 35649 specify each distinctly.

35650 Some historical implementations use standard input to get and set terminal characteristics;  
 35651 others use standard output. Since input from a login TTY is usually restricted to the owner while  
 35652 output to a TTY is frequently open to anyone, using standard input provides fewer chances of  
 35653 accidentally (or maliciously) altering the terminal settings of other users. Using standard input  
 35654 also allows *stty -a* and *stty -g* output to be redirected for later use. Therefore, usage of standard  
 35655 input is required by this volume of IEEE Std. 1003.1-200x.

35656 The **tostop** option is the only option that requires job control to be effective, and thus could have  
 35657 gone into the UPE as a modification to *stty*, but since all other terminal control features are in the  
 35658 base standard, **tostop** was included as well.

35659 **FUTURE DIRECTIONS**

35660 None.

35661 **SEE ALSO**35662 The System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal  
35663 Interface35664 **CHANGE HISTORY**

35665 First released in Issue 2.

35666 **Issue 4**

35667 Aligned with the ISO/IEC 9945-2: 1993 standard.

35668 **Issue 5**35669 The description of **tabs** is clarified.

35670 FUTURE DIRECTIONS section added.

35671 **Issue 6**35672 The legacy items **iuclt(-iuclt)**, **xcase**, **olcuc(-olcuc)**, **lcase(-lcase)**, and **LCASE(-LCASE)**, are  
35673 removed.

## 35674 NAME

35675 tabs — set terminal tabs

## 35676 SYNOPSIS

35677 UP XSI tabs [ -n | -a | -a2 | -c | -c2 | -c3 | -f | -p | -s | -u ][ +m[n] ] [-T type]

35678 tabs [-T type][ +[n] ] n1[,n2,...]

35679

## 35680 DESCRIPTION

35681 The *tabs* utility shall display a series of characters that first clears the hardware terminal tab  
 35682 XSI settings and then initializes the tab stops at the specified positions and optionally adjusts the  
 35683 margin.

35684 The phrase “tab-stop position *N*” shall be taken to mean that, from the start of a line of output,  
 35685 tabbing to position *N* shall cause the next character output to be in the (*N*+1)th column position  
 35686 on that line. The maximum number of tab stops allowed is terminal-dependent.

35687 It need not be possible to implement *tabs* on certain terminals. If the terminal type obtained from  
 35688 the *TERM* environment variable or *-T* option represents such a terminal, an appropriate  
 35689 diagnostic message shall be written to standard error and *tabs* shall exit with a status greater  
 35690 than zero.

## 35691 OPTIONS

35692 The *tabs* utility shall conform to the System Interface Definitions volume of  
 35693 XSI IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines, except for various extensions: the  
 35694 options *-a2*, *-c2*, and *-c3* are multi-character and *+m [n]* uses a leading plus sign and an  
 35695 optional option-argument.

35696 The following options shall be supported:

35697 *-n* Specify repetitive tab stops separated by a uniform number of column positions, *n*,  
 35698 where *n* is a single-digit decimal number. The default usage of *tabs* with no  
 35699 arguments shall be equivalent to *tabs-8*. When *-0* is used, the tab stops shall be  
 35700 cleared and no new ones set.

35701 XSI *-a* 1,10,16,36,72  
 35702 Assembler, applicable to some mainframes.

35703 XSI *-a2* 1,10,16,40,72  
 35704 Assembler, applicable to some mainframes.

35705 XSI *-c* 1,8,12,16,20,55  
 35706 COBOL, normal format.

35707 XSI *-c2* 1,6,10,14,49  
 35708 COBOL, compact format (columns 1-6 omitted).

35709 XSI *-c3* 1,6,10,14,18,22,26,30,34,38,42,46,50,54,58,62,67  
 35710 COBOL compact format (columns 1-6 omitted), with more tabs than *-c2*.

35711 XSI *-f* 1,7,11,15,19,23  
 35712 FORTRAN

35713 XSI *-p* 1,5,9,13,17,21,25,29,33,37,41,45,49,53,57,61  
 35714 PL/1

35715 XSI *-s* 1,10,55  
 35716 SNOBOL

|              |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|--------------|---------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 35717 XSI    | <b>-u</b>                                                                       | 1,12,20,44                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 35718        |                                                                                 | Assembler, applicable to some mainframes.                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 35719        | <b>-T type</b>                                                                  | Indicate the type of terminal. If this option is not supplied and the <i>TERM</i> variable is unset or null, an unspecified default terminal type shall be used. The setting of <i>type</i> shall take precedence over the value in <i>TERM</i> .                                                                                                                                                                                                            |
| 35720        |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35721        |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35722 XSI UN | <b>+m[n]</b>                                                                    | Reset the margin. The margin argument can be used for some terminals. It shall cause all tabs to be moved over <i>n</i> columns by making column <i>n</i> +1 the left margin. If <i>n</i> is omitted, the default shall be 10. The normal (leftmost) margin on most terminals is obtained by <b>+m0</b> . The margin for most terminals is reset only when the <b>+m</b> flag is given explicitly.                                                           |
| 35723        |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35724        |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35725        |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35726        |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35727        | <b>OPERANDS</b>                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35728        | The following operand shall be supported:                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35729        | <b>n1[,n2,...]</b>                                                              | A single command line argument that consists of tab-stop values separated using either commas or <blank> characters. The application shall ensure that the tab-stop values are positive decimal integers in strictly ascending order. If any number (except the first one) is preceded by a plus sign, it is taken as an increment to be added to the previous value. For example, the tab lists 1,10,20,30 and 1,10,+10,+10 are considered to be identical. |
| 35730        |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35731        |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35732        |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35733        |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35734        |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35735        | <b>STDIN</b>                                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35736        | Not used.                                                                       |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35737        | <b>INPUT FILES</b>                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35738        | None.                                                                           |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35739        | <b>ENVIRONMENT VARIABLES</b>                                                    |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35740        | The following environment variables shall affect the execution of <i>tabs</i> : |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35741        | <b>LANG</b>                                                                     | Provide a default value for the internationalization variables that are unset or null. If <i>LANG</i> is unset or null, the corresponding value from the implementation-dependent default locale shall be used. If any of the internationalization variables contains an invalid setting, the utility shall behave as if none of the variables had been defined.                                                                                             |
| 35742        |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35743        |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35744        |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35745        |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35746        | <b>LC_ALL</b>                                                                   | If set to a non-empty string value, override the values of all the other internationalization variables.                                                                                                                                                                                                                                                                                                                                                     |
| 35747        |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35748        | <b>LC_CTYPE</b>                                                                 | Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).                                                                                                                                                                                                                                                                                    |
| 35749        |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35750        |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35751        | <b>LC_MESSAGES</b>                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35752        |                                                                                 | Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.                                                                                                                                                                                                                                                                                                                                 |
| 35753        |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35754 XSI    | <b>NLSPATH</b>                                                                  | Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .                                                                                                                                                                                                                                                                                                                                                                        |
| 35755        | <b>TERM</b>                                                                     | Determine the terminal type. If this variable is unset or null, and if the <b>-T</b> option is not specified, an unspecified default terminal type shall be used.                                                                                                                                                                                                                                                                                            |
| 35756        |                                                                                 |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35757        | <b>ASYNCHRONOUS EVENTS</b>                                                      |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 35758        | Default.                                                                        |                                                                                                                                                                                                                                                                                                                                                                                                                                                              |



**35759 STDOUT**

35760 If standard output is a terminal, the appropriate sequence to clear and set the tab stops may be  
 35761 written to standard output in an unspecified format. If standard output is not a terminal,  
 35762 undefined results occur.

**35763 STDERR**

35764 Used only for diagnostic messages.

**35765 OUTPUT FILES**

35766 None.

**35767 EXTENDED DESCRIPTION**

35768 None.

**35769 EXIT STATUS**

35770 The following exit values shall be returned:

35771 0 Successful completion.

35772 >0 An error occurred.

**35773 CONSEQUENCES OF ERRORS**

35774 Default.

**35775 APPLICATION USAGE**

35776 This utility makes use of the terminal's hardware tabs and the *stty tabs* option.

35777 This utility is not recommended for application use.

35778 Some integrated display units might not have escape sequences to set tab stops, but may be set  
 35779 by internal system calls. On these terminals, *tabs* works if standard output is directed to the  
 35780 terminal; if output is directed to another file, however, *tabs* fails.

35781 Application writers should note that this utility need not be provided on systems that do not  
 35782 support the User Portability Utilities option.

**35783 EXAMPLES**

35784 None.

**35785 RATIONALE**

35786 Consideration was given to having the *tput* utility handle all of the functions described in *tabs*.  
 35787 However, the separate *tabs* utility was retained because it seems more intuitive to use a  
 35788 command named *tabs* than *tput* with a new option. The POSIX Shell and Utilities *tput* does not  
 35789 support setting or clearing tabs, and no known historical version of *tabs* supports the capability  
 35790 of setting arbitrary tab stops.

35791 The System V *tabs* interface is very complex; the version in this volume of IEEE Std. 1003.1-200x  
 35792 has a reduced feature list. There was considerable sentiment for specifying only a means of  
 35793 resetting the tabs back to a known state—presumably the “standard” of tabs every eight  
 35794 positions. The following features were omitted:

- 35795 • Setting tab stops tailored for certain programming languages; the standard developers were  
 35796 concerned that it would be difficult to decide which languages to include and where the tabs  
 35797 should be.
- 35798 • Setting tab stops via the first line in a file, using *—file*. Since even the SVID has no complete  
 35799 explanation of this feature, it is doubtful that it is in widespread use.
- 35800 • Setting the left margin using *+mn*. As this does not work with all terminal types, it was  
 35801 omitted.

35802 In an early proposal, a **-t** *tablist* option was added for consistency with *expand*; this was later  
 35803 removed when inconsistencies with the historical list of tabs were identified.

35804 Consideration was given to adding a **-p** option that would output the current tab settings so  
 35805 that they could be saved and then later restored. This was not accepted because querying the tab  
 35806 stops of the terminal is not a capability in historical *terminfo* or *termcap* facilities and might not be  
 35807 supported on a wide range of terminals.

35808 **FUTURE DIRECTIONS**

35809 None.

35810 **SEE ALSO**

35811 *expand, stty, unexpand*

35812 **CHANGE HISTORY**

35813 First released in Issue 2.

35814 **Issue 4**

35815 Aligned with the ISO/IEC 9945-2: 1993 standard.

35816 **Issue 6**

35817 This utility is now marked as part of the User Portability Utilities option. |

35818 The normative text is reworded to avoid use of the term “must” for application requirements. |

35819 **NAME**35820 **tail** — copy the last part of a file35821 **SYNOPSIS**35822 **tail** [-f][ -c *number* | -n *number*][*file*]35823 **DESCRIPTION**35824 The *tail* utility shall copy its input file to the standard output beginning at a designated place.

35825 Copying shall begin at the point in the file indicated by the **-c** *number* or **-n** *number* options. The  
 35826 option-argument *number* shall be counted in units of lines or bytes, according to the options **-n**  
 35827 and **-c**. Both line and byte counts start from 1.

35828 Tails relative to the end of the file may be saved in an internal buffer, and thus may be limited in  
 35829 length. Such a buffer, if any, is no smaller than {LINE\_MAX}\*10 bytes.

35830 **OPTIONS**

35831 The *tail* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 35832 Section 12.2, Utility Syntax Guidelines.

35833 The following options shall be supported:

35834 **-c** *number* The application shall ensure that the *number* option-argument is a decimal integer  
 35835 whose sign affects the location in the file, measured in bytes, to begin the copying:

| 35836 | Sign        | Copying Starts                         |
|-------|-------------|----------------------------------------|
| 35837 | +           | Relative to the beginning of the file. |
| 35838 | -           | Relative to the end of the file.       |
| 35839 | <i>none</i> | Relative to the end of the file.       |

35840 The origin for counting shall be 1; that is, **-c** +1 represents the first byte of the file,  
 35841 **-c** -1 the last.

35842 **-f** If the input file is a regular file or if the *file* operand specifies a FIFO, do not  
 35843 terminate after the last line of the input file has been copied, but read and copy  
 35844 further bytes from the input file when they become available. If no *file* operand is  
 35845 specified and standard input is a pipe, the **-f** option shall be ignored. If the input  
 35846 file is not a FIFO, pipe, or regular file, it is unspecified whether or not the **-f** option  
 35847 shall be ignored.

35848 **-n** *number* This option is equivalent to **-c** *number*, except the starting location in the file shall  
 35849 be measured in lines instead of bytes. The origin for counting shall be 1; that is, **-n**  
 35850 +1 represents the first line of the file, **-n** -1 the last.

35851 If neither **-c** nor **-n** is specified, **-n** 10 shall be assumed.35852 **OPERANDS**

35853 The following operand shall be supported:

35854 *file* A path name of an input file. If no *file* operands are specified, the standard input  
 35855 shall be used.

35856 **STDIN**

35857 The standard input shall be used only if no *file* operands are specified. See the INPUT FILES  
 35858 section.

35859 **INPUT FILES**

35860 If the **-c** option is specified, the input file can contain arbitrary data; otherwise, the input file  
 35861 shall be a text file.

35862 **ENVIRONMENT VARIABLES**

35863 The following environment variables shall affect the execution of *tail*:

35864 **LANG** Provide a default value for the internationalization variables that are unset or null.  
 35865 If **LANG** is unset or null, the corresponding value from the implementation-  
 35866 dependent default locale shall be used. If any of the internationalization variables  
 35867 contains an invalid setting, the utility shall behave as if none of the variables had  
 35868 been defined.

35869 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 35870 internationalization variables.

35871 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 35872 characters (for example, single-byte as opposed to multi-byte characters in  
 35873 arguments and input files).

35874 **LC\_MESSAGES**  
 35875 Determine the locale that should be used to affect the format and contents of  
 35876 diagnostic messages written to standard error.

35877 **XS** **NLSPATH** Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

35878 **ASYNCHRONOUS EVENTS**

35879 Default.

35880 **STDOUT**

35881 The designated portion of the input file shall be written to standard output.

35882 **STDERR**

35883 Used only for diagnostic messages.

35884 **OUTPUT FILES**

35885 None.

35886 **EXTENDED DESCRIPTION**

35887 None.

35888 **EXIT STATUS**

35889 The following exit values shall be returned:

35890 **0** Successful completion.

35891 **>0** An error occurred.

35892 **CONSEQUENCES OF ERRORS**

35893 Default.

**35894 APPLICATION USAGE**

35895 The **-c** option should be used with caution when the input is a text file containing multi-byte  
35896 characters; it may produce output that does not start on a character boundary.

35897 Although the input file to *tail* can be any type, the results might not be what would be expected  
35898 on some character special device files or on file types not described by the System Interfaces  
35899 volume of IEEE Std. 1003.1-200x. Since this volume of IEEE Std. 1003.1-200x does not specify the  
35900 block size used when doing input, *tail* need not read all of the data from devices that only  
35901 perform block transfers.

**35902 EXAMPLES**

35903 The **-f** option can be used to monitor the growth of a file that is being written by some other  
35904 process. For example, the command:

35905 `tail -f fred`

35906 prints the last ten lines of the file **fred**, followed by any lines that are appended to **fred** between  
35907 the time *tail* is initiated and killed. As another example, the command:

35908 `tail -f -c 15 fred`

35909 prints the last 15 bytes of the file **fred**, followed by any bytes that are appended to **fred** between  
35910 the time *tail* is initiated and killed.

**35911 RATIONALE**

35912 This version of *tail* was created to allow conformance to the Utility Syntax Guidelines. The  
35913 historical **-b** option was omitted because of the general non-portability of block-sized units of  
35914 text. The **-c** option historically meant “characters”, but this volume of IEEE Std. 1003.1-200x  
35915 indicates that it means “bytes”. This was selected to allow reasonable implementations when  
35916 multi-byte characters are possible; it was not named **-b** to avoid confusion with the historical  
35917 **-b**.

35918 The origin of counting both lines and bytes is 1, matching all widespread historical  
35919 implementations.

35920 The restriction on the internal buffer is a compromise between the historical System V  
35921 implementation of 4 096 bytes and the BSD 32 768 bytes.

35922 The **-f** option has been implemented as a loop that sleeps for 1 second and copies any bytes that  
35923 are available. This is sufficient, but if more efficient methods of determining when new data are  
35924 available are developed, implementations are encouraged to use them.

35925 Historical documentation indicates that *tail* ignores the **-f** option if the input file is a pipe (pipe  
35926 and FIFO on systems that support FIFOs). On BSD-based systems, this has been true; on System  
35927 V-based systems, this was true when input was taken from standard input, but it did not ignore  
35928 the **-f** flag if a FIFO was named as the *file* operand. Since the **-f** option is not useful on pipes and  
35929 all historical implementations ignore **-f** if no *file* operand is specified and standard input is a  
35930 pipe, this volume of IEEE Std. 1003.1-200x requires this behavior. However, since the **-f** option is  
35931 useful on a FIFO, this volume of IEEE Std. 1003.1-200x also requires that if standard input is a  
35932 FIFO or a FIFO is named, the **-f** option shall not be ignored. Although historical behavior does  
35933 not ignore the **-f** option for other file types, this is unspecified so that implementations are  
35934 allowed to ignore the **-f** option if it is known that the file cannot be extended.

35935 This was changed to the current form based on comments noting that **-c** was almost never used  
35936 without specifying a number and that there was no need to specify **-l** if **-n number** was given.

|       |                                                                                              |  |
|-------|----------------------------------------------------------------------------------------------|--|
| 35937 | <b>FUTURE DIRECTIONS</b>                                                                     |  |
| 35938 | None.                                                                                        |  |
| 35939 | <b>SEE ALSO</b>                                                                              |  |
| 35940 | <i>head</i>                                                                                  |  |
| 35941 | <b>CHANGE HISTORY</b>                                                                        |  |
| 35942 | First released in Issue 2.                                                                   |  |
| 35943 | <b>Issue 4</b>                                                                               |  |
| 35944 | Aligned with the ISO/IEC 9945-2: 1993 standard.                                              |  |
| 35945 | <b>Issue 6</b>                                                                               |  |
| 35946 | The obsolescent SYNOPSIS lines and associated text are removed.                              |  |
| 35947 | The normative text is reworded to avoid use of the term “must” for application requirements. |  |

35948 **NAME**35949           *talk* — talk to another user35950 **SYNOPSIS**35951 UP       *talk address [terminal]*

35952

35953 **DESCRIPTION**35954       The *talk* utility is a two-way, screen-oriented communication program.35955       When first invoked, *talk* shall send a message similar to:

35956       Message from <*unspecified string*>  
 35957       *talk*: connection requested by *your\_address*  
 35958       *talk*: respond with: *talk your\_address*

35959       to the specified *address*. At this point, the recipient of the message can reply by typing:35960       *talk your\_address*

35961       Once communication is established, the two parties can type simultaneously, with their output  
 35962       displayed in separate regions of the screen. Characters shall be processed as follows:

- 35963       • Typing the alert character shall alert the recipient's terminal.
- 35964       • Typing <control>-L shall cause the sender's screen regions to be refreshed.
- 35965       • Typing the erase and kill characters shall affect the sender's terminal in the manner described  
 35966       by the **termios** interface in the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 35967       Chapter 11, General Terminal Interface.
- 35968       • Typing the interrupt or end-of-file characters shall terminate the local *talk* utility. Once the  
 35969       *talk* session has been terminated on one side, the other side of the *talk* session shall be notified  
 35970       that the *talk* session has been terminated and shall be able to do nothing except exit.
- 35971       • Typing characters from *LC\_CTYPE* classifications **print** or **space** shall cause those characters  
 35972       to be sent to the recipient's terminal.
- 35973       • When and only when the *stty iexten* local mode is enabled, the existence and processing of  
 35974       additional special control characters and multi-byte or single-byte functions shall be  
 35975       implementation-dependent.
- 35976       • Typing other non-printable characters shall cause implementation-dependent sequences of  
 35977       printable characters to be sent to the recipient's terminal.

35978       Permission to be a recipient of a *talk* message can be denied or granted by use of the *mesg* utility.  
 35979       However, a user's privilege may further constrain the domain of accessibility of other users'  
 35980       terminals. The *talk* utility shall fail when the user lacks the appropriate privileges to perform the  
 35981       requested action.

35982       Certain block-mode terminals do not have all the capabilities necessary to support the  
 35983       simultaneous exchange of messages required for *talk*. When this type of exchange cannot be  
 35984       supported on such terminals, the implementation may support an exchange with reduced levels  
 35985       of simultaneous interaction or it may report an error describing the terminal-related deficiency.

35986 **OPTIONS**

35987       None.

35988 **OPERANDS**

35989           The following operands shall be supported:

35990           *address*       The recipient of the *talk* session. One form of *address* is the <*user name*>, as returned  
35991                           by the *who* utility. Other address formats and how they are handled are  
35992                           unspecified.

35993           *terminal*      If the recipient is logged in more than once, the *terminal* argument can be used to  
35994                           indicate the appropriate terminal name. If *terminal* is not specified, the *talk* message  
35995                           shall be displayed on one or more accessible terminals in use by the recipient. The  
35996                           format of *terminal* shall be the same as that returned by the *who* utility.

35997 **STDIN**

35998           Characters read from standard input shall be copied to the recipient's terminal in an unspecified  
35999           manner. If standard input is not a terminal, *talk* shall write a diagnostic message and exit with a  
36000           non-zero status.

36001 **INPUT FILES**

36002           None.

36003 **ENVIRONMENT VARIABLES**

36004           The following environment variables shall affect the execution of *talk*:

36005           *LANG*           Provide a default value for the internationalization variables that are unset or null.  
36006                           If *LANG* is unset or null, the corresponding value from the implementation-  
36007                           dependent default locale shall be used. If any of the internationalization variables  
36008                           contains an invalid setting, the utility shall behave as if none of the variables had  
36009                           been defined.

36010           *LC\_ALL*       If set to a non-empty string value, override the values of all the other  
36011                           internationalization variables.

36012           *LC\_CTYPE*   Determine the locale for the interpretation of sequences of bytes of text data as  
36013                           characters (for example, single-byte as opposed to multi-byte characters in  
36014                           arguments and input files). If the recipient's locale does not use an *LC\_CTYPE*  
36015                           equivalent to the sender's, the results are undefined.

36016 *LC\_MESSAGES*

36017                           Determine the locale that should be used to affect the format and contents of  
36018                           diagnostic messages written to standard error and informative messages written to  
36019                           standard output.

36020 XSI       *NLSPATH*   Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

36021           *TERM*           Determine the name of the invoker's terminal type. If this variable is unset or null,  
36022                           an unspecified default terminal type shall be used.

36023 **ASYNCHRONOUS EVENTS**

36024           When the *talk* utility receives a SIGINT signal, the utility shall terminate and exit with a zero  
36025           status. It shall take the standard action for all other signals.

36026 **STDOUT**

36027           If standard output is a terminal, characters copied from the recipient's standard input may be  
36028           written to standard output. Standard output also may be used for diagnostic messages. If  
36029           standard output is not a terminal, *talk* shall exit with a non-zero status.



36030 **STDERR**

36031 None.

36032 **OUTPUT FILES**

36033 None.

36034 **EXTENDED DESCRIPTION**

36035 None.

36036 **EXIT STATUS**

36037 The following exit values shall be returned:

36038 0 Successful completion.

36039 >0 An error occurred or *talk* was invoked on a terminal incapable of supporting it.36040 **CONSEQUENCES OF ERRORS**

36041 Default.

36042 **APPLICATION USAGE**

36043 Because the handling of non-printable, non-`<space>` characters is tied to the *stty* description of  
 36044 **ixten**, implementation extensions within the terminal driver can be accessed. For example,  
 36045 some implementations provide line editing functions with certain control character sequences.

36046 Application writers should note that this utility need not be provided on systems that do not  
 36047 support the User Portability Utilities option.

36048 **EXAMPLES**

36049 None.

36050 **RATIONALE**

36051 The *write* utility was included in this volume of IEEE Std. 1003.1-200x since it can be  
 36052 implemented on all terminal types. The *talk* utility, which cannot be implemented on certain  
 36053 terminals, was considered to be a “better” communications interface. Both of these programs are  
 36054 in widespread use on historical implementations. Therefore, both utilities have been specified.

36055 All references to networking abilities (*talking* to a user on another system) were removed as  
 36056 being outside the scope of this volume of IEEE Std. 1003.1-200x.

36057 Historical BSD and System V versions of *talk* terminate both of the conversations when either  
 36058 user breaks out of the session. This can lead to adverse consequences if a user unwittingly  
 36059 continues to enter text that is interpreted by the shell when the other terminates the session.  
 36060 Therefore, the version of *talk* specified by this volume of IEEE Std. 1003.1-200x requires both  
 36061 users to terminate their end of the session explicitly.

36062 Only messages sent to the terminal of the invoking user can be internationalized in any way:

- 36063 • The original “Message from *<unspecified string>* ...” message sent to the terminal of the  
 36064 recipient cannot be internationalized because the environment of the recipient is as yet  
 36065 inaccessible to the *talk* utility. The environment of the invoking party is irrelevant.
- 36066 • Subsequent communication between the two parties cannot be internationalized because the  
 36067 two parties may specify different languages in their environment (and non-portable  
 36068 characters cannot be mapped from one language to another).
- 36069 • Neither party can be required to communicate in a language other than C and/or the one  
 36070 specified by their environment because unavailable terminal hardware support (for example,  
 36071 fonts) may be required.

36072 The text in the STDOUT section reflects the usage of the verb “display” in this section; some *talk*  
 36073 implementations actually use standard output to write to the terminal, but this volume of

- 36074 IEEE Std. 1003.1-200x does not require that to be the case.
- 36075 The format of the terminal name is unspecified, but the descriptions of *ps*, *talk*, *who*, and *write*  
36076 require that they all use or accept the same format.
- 36077 The handling of non-printable characters is partially implementation-dependent because the  
36078 details of mapping them to printable sequences is not needed by the user. Historical  
36079 implementations, for security reasons, disallow the transmission of non-printable characters that  
36080 may send commands to the other terminal.
- 36081 **FUTURE DIRECTIONS**
- 36082 None.
- 36083 **SEE ALSO**
- 36084 *mesg*, *who*, *write*, the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, |  
36085 General Terminal Interface |
- 36086 **CHANGE HISTORY**
- 36087 First released in Issue 4.
- 36088 **Issue 6**
- 36089 This utility is now marked as part of the User Portability Utilities option.

36090 **NAME**

36091           tee — duplicate standard input

36092 **SYNOPSIS**36093           tee [-ai][*file...*]36094 **DESCRIPTION**36095           The *tee* utility shall copy standard input to standard output, making a copy in zero or more files.36096           The *tee* utility shall not buffer output.

36097           The options determine whether the specified files are overwritten or appended to.

36098 **OPTIONS**36099           The *tee* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
36100           Section 12.2, Utility Syntax Guidelines.

36101           The following options shall be supported:

36102           -a           Append the output to the files rather than overwriting them.

36103           -i           Ignore the SIGINT signal.

36104 **OPERANDS**

36105           The following operands shall be supported:

36106           *file*        A path name of an output file. Processing of at least 13 *file* operands shall be  
36107           supported.36108 **STDIN**

36109           The standard input can be of any type.

36110 **INPUT FILES**

36111           None.

36112 **ENVIRONMENT VARIABLES**36113           The following environment variables shall affect the execution of *tee*:36114           *LANG*        Provide a default value for the internationalization variables that are unset or null.  
36115                            If *LANG* is unset or null, the corresponding value from the implementation-  
36116                            dependent default locale shall be used. If any of the internationalization variables  
36117                            contains an invalid setting, the utility shall behave as if none of the variables had  
36118                            been defined.36119           *LC\_ALL*       If set to a non-empty string value, override the values of all the other  
36120                            internationalization variables.36121           *LC\_CTYPE*   Determine the locale for the interpretation of sequences of bytes of text data as  
36122                            characters (for example, single-byte as opposed to multi-byte characters in  
36123                            arguments).36124           *LC\_MESSAGES*36125                           Determine the locale that should be used to affect the format and contents of  
36126                            diagnostic messages written to standard error.36127 XSI       *NLS\_PATH*   Determine the location of message catalogs for the processing of *LC\_MESSAGES*.36128 **ASYNCHRONOUS EVENTS**

36129           Default, except that if the -i option was specified, SIGINT shall be ignored.

**36130 STDOUT**

36131 The standard output shall be a copy of the standard input.

**36132 STDERR**

36133 Used only for diagnostic messages.

**36134 OUTPUT FILES**

36135 If any *file* operands are specified, the standard input shall be copied to each named file.

**36136 EXTENDED DESCRIPTION**

36137 None.

**36138 EXIT STATUS**

36139 The following exit values shall be returned:

36140 0 The standard input was successfully copied to all output files.

36141 >0 An error occurred.

**36142 CONSEQUENCES OF ERRORS**

36143 If a write to any successfully opened *file* operand fails, writes to other successfully opened *file*  
36144 operands and standard output shall continue, but the exit status shall be non-zero. Otherwise,  
36145 the default actions specified in Section 1.11 on page 25 apply.

**36146 APPLICATION USAGE**

36147 The *tee* utility is usually used in a pipeline, to make a copy of the output of some utility.

36148 The *file* operand is technically optional, but *tee* is no more useful than *cat* when none is specified.

**36149 EXAMPLES**

36150 Save an unsorted intermediate form of the data in a pipeline:

36151 ... | tee unsorted | sort > sorted

**36152 RATIONALE**

36153 The buffering requirement means that *tee* is not allowed to use ISO C standard fully buffered or  
36154 line-buffered writes. It does not mean that *tee* has to do 1-byte reads followed by 1-byte writes.

36155 It should be noted that early versions of BSD ignore any invalid options and accept a single '-'  
36156 as an alternative to -i. They also print a message if unable to open a file:

36157 "tee: cannot access %s\n", <pathname>

36158 Historical implementations ignore write errors. This is explicitly not permitted by this volume of  
36159 IEEE Std. 1003.1-200x.

36160 Some historical implementations use O\_APPEND when providing append mode; others use the  
36161 *lseek()* function to seek to the end of file after opening the file without O\_APPEND. This volume  
36162 of IEEE Std. 1003.1-200x requires functionality equivalent to using O\_APPEND; see Section  
36163 1.7.1.4 on page 11.

**36164 FUTURE DIRECTIONS**

36165 None.

**36166 SEE ALSO**

36167 *cat*

**36168 CHANGE HISTORY**

36169 First released in Issue 2.

36170 **Issue 4**

36171 Aligned with the ISO/IEC 9945-2: 1993 standard.

36172 **NAME**

36173       test — evaluate expression

36174 **SYNOPSIS**36175       test [*expression*]36176       [ [*expression*] ]36177 **DESCRIPTION**

36178       The *test* utility shall evaluate the *expression* and indicates the result of the evaluation by its exit  
 36179       status. An exit status of zero indicates that the expression evaluated as true and an exit status of  
 36180       1 indicates that the expression evaluated as false.

36181       In the second form of the utility, which uses "[ ]" rather than *test*, the application shall ensure  
 36182       that the square brackets are separate arguments.

36183 **OPTIONS**

36184       The *test* utility shall not recognize the "—" argument in the manner specified by guideline 10 in  
 36185       the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax  
 36186       Guidelines.

36187       No options shall be supported.

36188 **OPERANDS**

36189       The application shall ensure that all operators and elements of primaries are presented as  
 36190       separate arguments to the *test* utility.

36191       The following primaries can be used to construct *expression*:

36192       **-b file**       True if *file* exists and is a block special file.

36193       **-c file**       True if *file* exists and is a character special file.

36194       **-d file**       True if *file* exists and is a directory.

36195       **-e file**       True if *file* exists.

36196       **-f file**       True if *file* exists and is a regular file.

36197       **-g file**       True if *file* exists and its set group ID flag is set.

36198       **-h file**       True if *file* exists and is a symbolic link.

36199       **-n string**     True if the length of *string* is non-zero.

36200       **-p file**       True if *file* is a named pipe (FIFO).

36201       **-r file**       True if *file* exists and is readable. True shall indicate that permission to read from  
 36202       *file* will be granted, as defined in Section 1.7.1.4 on page 11.

36203       **-s file**       True if *file* exists and has a size greater than zero.

36204       **-t file\_descriptor**

36205       True if the file whose file descriptor number is *file\_descriptor* is open and is  
 36206       associated with a terminal.

36207       **-u file**       True if *file* exists and its set-user-ID flag is set.

36208       **-w file**       True if *file* exists and is writable. True shall indicate that permission to write from  
 36209       *file* will be granted, as defined in Section 1.7.1.4 on page 11.

36210       **-x file**       True if *file* exists and is executable. True if *file* exists and is executable. True shall  
 36211       indicate that permission to execute *file* will be granted, as defined in Section 1.7.1.4  
 36212       on page 11. If *file* is a directory, true shall indicate that permission to search *file*

36213

will be granted.

36214

-z string

True if the length of string *string* is zero.

36215

string

True if the string *string* is not the null string.

36216

s1 = s2

True if the strings *s1* and *s2* are identical.

36217

s1 != s2

True if the strings *s1* and *s2* are not identical.

36218

n1 -eq n2

True if the integers *n1* and *n2* are algebraically equal.

36219

n1 -ne n2

True if the integers *n1* and *n2* are not algebraically equal.

36220

n1 -gt n2

True if the integer *n1* is algebraically greater than the integer *n2*.

36221

n1 -ge n2

True if the integer *n1* is algebraically greater than or equal to the integer *n2*.

36222

n1 -lt n2

True if the integer *n1* is algebraically less than the integer *n2*.

36223

n1 -le n2

True if the integer *n1* is algebraically less than or equal to the integer *n2*.

36224 XSI

expression1 -a expression2

True if both *expression1* and *expression2* are true. The -a binary primary is left

36225

associative. It has a higher precedence than -o.

36226

36227 XSI

expression1 -o expression2

True if either *expression1* or *expression2* is true. The -o binary primary is left

36228

associative.

36229

36230

With the exception of the -h file primary, if a file argument is a symbolic link, test shall evaluate

36231

the expression by resolving the symbolic link and using the file referenced by the link.

36232

These primaries can be combined with the following operators:

36233

! expression

True if *expression* is false.

36234 XSI

( expression )

True if *expression* is true. The parentheses can be used to alter the normal

36235

precedence and associativity.

36236

The primaries with two elements of the form:

36237

-primary\_operator primary\_operand

36238

are known as unary primaries. The primaries with three elements in either of the two forms:

36239

primary\_operand -primary\_operator primary\_operand

36240

primary\_operand primary\_operator primary\_operand

36241

are known as binary primaries. Additional implementation-dependent operators and

36242

primary\_operators may be provided by implementations. They shall be of the form -operator

36243

where the first character of operator is not a digit.

36244

The algorithm for determining the precedence of the operators and the return value that shall be

36245

generated is based on the number of arguments presented to test. (However, when using the

36246

" [ . . . ] " form, the right-bracket final argument shall not be counted in this algorithm.)

36247

In the following list, \$1, \$2, \$3, and \$4 represent the arguments presented to test:

36248

0 arguments:

Exit false (1).

36249

1 argument:

Exit true (0) if \$1 is not null; otherwise, exit false.

36250

2 arguments:

• If \$1 is '!', exit true if \$2 is null, false if \$2 is not null.

|       |                              |                                                                                                                                                                                                                                                                                                                                                                  |
|-------|------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 36251 |                              | • If \$1 is a unary primary, exit true if the unary test is true, false if the unary test is false.                                                                                                                                                                                                                                                              |
| 36252 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 36253 |                              | • Otherwise, produce unspecified results.                                                                                                                                                                                                                                                                                                                        |
| 36254 | 3 arguments:                 | • If \$2 is a binary primary, perform the binary test of \$1 and \$3.                                                                                                                                                                                                                                                                                            |
| 36255 |                              | • If \$1 is ' ! ', negate the two-argument test of \$2 and \$3.                                                                                                                                                                                                                                                                                                  |
| 36256 | MAN                          | • If \$1 is ' ( ' and \$3 is ' ) ', perform the unary test of \$2.                                                                                                                                                                                                                                                                                               |
| 36257 |                              | • Otherwise, produce unspecified results.                                                                                                                                                                                                                                                                                                                        |
| 36258 | 4 arguments:                 | • If \$1 is ' ! ', negate the three-argument test of \$2, \$3, and \$4.                                                                                                                                                                                                                                                                                          |
| 36259 | XSI                          | • If \$1 is ' ( ' and \$4 is ' ) ', perform the two-argument test of \$2 and \$3.                                                                                                                                                                                                                                                                                |
| 36260 |                              | • Otherwise, the results are unspecified.                                                                                                                                                                                                                                                                                                                        |
| 36261 | XSI                          | >4 arguments: Combinations of primaries and operators shall be evaluated using the precedence and associativity rules described previously. In addition, the string comparison binary primaries '=' and '!=' shall have a higher precedence than any unary primary.                                                                                              |
| 36262 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 36263 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 36264 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 36265 | <b>STDIN</b>                 |                                                                                                                                                                                                                                                                                                                                                                  |
| 36266 |                              | Not used.                                                                                                                                                                                                                                                                                                                                                        |
| 36267 | <b>INPUT FILES</b>           |                                                                                                                                                                                                                                                                                                                                                                  |
| 36268 |                              | None.                                                                                                                                                                                                                                                                                                                                                            |
| 36269 | <b>ENVIRONMENT VARIABLES</b> |                                                                                                                                                                                                                                                                                                                                                                  |
| 36270 |                              | The following environment variables shall affect the execution of <i>test</i> :                                                                                                                                                                                                                                                                                  |
| 36271 | <i>LANG</i>                  | Provide a default value for the internationalization variables that are unset or null. If <i>LANG</i> is unset or null, the corresponding value from the implementation-dependent default locale shall be used. If any of the internationalization variables contains an invalid setting, the utility shall behave as if none of the variables had been defined. |
| 36272 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 36273 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 36274 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 36275 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 36276 | <i>LC_ALL</i>                | If set to a non-empty string value, override the values of all the other internationalization variables.                                                                                                                                                                                                                                                         |
| 36277 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 36278 | <i>LC_CTYPE</i>              | Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments).                                                                                                                                                                                        |
| 36279 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 36280 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 36281 | <i>LC_MESSAGES</i>           | Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.                                                                                                                                                                                                                                     |
| 36282 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 36283 |                              |                                                                                                                                                                                                                                                                                                                                                                  |
| 36284 | XSI                          | <i>NLSPATH</i> Determine the location of message catalogs for the processing of <i>LC_MESSAGES</i> .                                                                                                                                                                                                                                                             |
| 36285 | <b>ASYNCHRONOUS EVENTS</b>   |                                                                                                                                                                                                                                                                                                                                                                  |
| 36286 |                              | Default.                                                                                                                                                                                                                                                                                                                                                         |
| 36287 | <b>STDOUT</b>                |                                                                                                                                                                                                                                                                                                                                                                  |
| 36288 |                              | Not used.                                                                                                                                                                                                                                                                                                                                                        |
| 36289 | <b>STDERR</b>                |                                                                                                                                                                                                                                                                                                                                                                  |
| 36290 |                              | Used only for diagnostic messages.                                                                                                                                                                                                                                                                                                                               |



36291 **OUTPUT FILES**

36292 None.

36293 **EXTENDED DESCRIPTION**

36294 None.

36295 **EXIT STATUS**

36296 The following exit values shall be returned:

36297 0 *expression* evaluated to true.36298 1 *expression* evaluated to false or *expression* was missing.

36299 &gt;1 An error occurred.

36300 **CONSEQUENCES OF ERRORS**

36301 Default.

36302 **APPLICATION USAGE**

36303 Scripts should be careful when dealing with user-supplied input that could be confused with  
 36304 primaries and operators. Unless the application writer knows all the cases that produce input to  
 36305 the script, invocations like:

36306 `test "$1" -a "$2"`

36307 should be written as:

36308 `test "$1" && test "$2"`

36309 to avoid problems if a user supplied values such as \$1 set to '!' and \$2 set to the null string.  
 36310 That is, in cases where maximal portability is of concern, replace:

36311 `test expr1 -a expr2`

36312 with:

36313 `test expr1 && test expr2`

36314 and replace:

36315 `test expr1 -o expr2`

36316 with:

36317 `test expr1 || test expr2`

36318 but note that, in *test*, `-a` has higher precedence than `-o` while `"&&"` and `"||"` have equal  
 36319 precedence in the shell.

36320 Parentheses or braces can be used in the shell command language to effect grouping.

36321 Parentheses must be escaped when using *sh*; for example:36322 `test \( expr1 -a expr2 \) -o expr3`

36323 This command is not always portable outside XSI-conformant systems. The following form can  
 36324 be used instead:

36325 `( test expr1 && test expr2 ) || test expr3`

36326 The two commands:

36327 `test "$1"`36328 `test ! "$1"`

36329 could not be used reliably on some historical systems. Unexpected results would occur if such a  
 36330 *string* expression were used and \$1 expanded to '!', '(', or a known unary primary. Better  
 36331 constructs are:

36332 test -n "\$1"  
 36333 test -z "\$1"

36334 respectively.

36335 Historical systems have also been unreliable given the common construct:

36336 test "\$response" = "expected string"

36337 One of the following is a more reliable form:

36338 test "X\$response" = "Xexpected string"  
 36339 test "expected string" = "\$response"

36340 Note that the second form assumes that *expected string* could not be confused with any unary  
 36341 primary. If *expected string* starts with '-', '(', '!', or even '=', the first form should be used  
 36342 instead. Using the preceding rules without the marked extensions, any of the three comparison  
 36343 forms is reliable, given any input. (However, note that the strings are quoted in all cases.)

36344 Because the string comparison binary primaries, '=' and '!=', have a higher precedence than  
 36345 any unary primary in the greater than 4 argument case, unexpected results can occur if  
 36346 arguments are not properly prepared. For example, in:

36347 test -d \$1 -o -d \$2

36348 If \$1 evaluates to a possible directory name of '=', the first three arguments are considered a  
 36349 string comparison, which shall cause a syntax error when the second -d is encountered. One of  
 36350 the following forms prevents this; the second is preferred:

36351 test \( -d "\$1" \) -o \( -d "\$2" \)  
 36352 test -d "\$1" || test -d "\$2"

36353 Also in the greater than 4 argument case:

36354 test "\$1" = "bat" -a "\$2" = "ball"

36355 Syntax errors occur if \$1 evaluates to '(' or '!'. One of the following forms prevents this; the  
 36356 third is preferred:

36357 test "X\$1" = "Xbat" -a "X\$2" = "Xball"  
 36358 test "\$1" = "bat" && test "\$2" = "ball"  
 36359 test "X\$1" = "Xbat" && test "X\$2" = "Xball"

## 36360 EXAMPLES

36361 1. Exit if there are not two or three arguments (two variations):

36362 if [ \$# -ne 2 -a \$# -ne 3 ]; then exit 1; fi  
 36363 if [ \$# -lt 2 -o \$# -gt 3 ]; then exit 1; fi

36364 2. Perform a *mkdir* if a directory does not exist:

36365 test ! -d tempdir && mkdir tempdir

36366 3. Wait for a file to become non-readable:

36367 while test -r thefile  
 36368 do  
 36369 sleep 30

```

36370 done
36371 echo '"thefile" is no longer readable'
36372
36373 4. Perform a command if the argument is one of three strings (two variations):
36374
36375 if ["$1" = "pear"] || ["$1" = "grape"] || ["$1" = "apple"]
36376 then
36377 command
36378 fi
36379
36380 case "$1" in
36381 pear|grape|apple) command ;;
36382 esac

```

#### 36380 RATIONALE

36381 The KornShell-derived conditional command (double bracket `[[ ]]`) was removed from the shell  
 36382 command language description in an early proposal. Objections were raised that the real  
 36383 problem is misuse of the `test` command (`I`), and putting it into the shell is the wrong way to fix  
 36384 the problem. Instead, proper documentation and a new shell reserved word (`!`) are sufficient.

36385 Tests that require multiple *test* operations can be done at the shell level using individual  
 36386 invocations of the *test* command and shell logicals, rather than using the error-prone `-o` flag of  
 36387 *test*.

36388 XSI-conformant systems support more than four arguments.

36389 XSI-conformant systems support the combining of primaries with the following constructs:

36390 *expression1* **-a** *expression2*  
 36391 True if both *expression1* and *expression2* are true.

36392 *expression1* **-o** *expression2*  
 36393 True if at least one of *expression1* and *expression2* are true.

36394 (*expression*)  
 36395 True if *expression* is true.

36396 In evaluating these more complex combined expressions, the following precedence rules are  
 36397 used:

- 36398 • The unary primaries have higher precedence than the algebraic binary primaries.
- 36399 • The unary primaries have lower precedence than the string binary primaries.
- 36400 • The unary and binary primaries have higher precedence than the unary *string* primary.
- 36401 • The `!` operator has higher precedence than the `-a` operator, and the `-a` operator has higher  
 36402 precedence than the `-o` operator.
- 36403 • The `-a` and `-o` operators are left associative.
- 36404 • The parentheses can be used to alter the normal precedence and associativity.

36405 The BSD and System V versions of `-f` are not the same. The BSD definition was:

36406 **-f** *file* True if *file* exists and is not a directory.

36407 The SVID version (true if the file exists and is a regular file) was chosen for this volume of  
 36408 IEEE Std. 1003.1-200x because its use is consistent with the `-b`, `-c`, `-d`, and `-p` operands (*file*  
 36409 exists and is a specific file type).

36410 The `-e` primary, possessing similar functionality to that provided by the C shell, was added  
 36411 because it provides the only way for a shell script to find out if a file exists without trying to

open the file. Since implementations are allowed to add additional file types, a portable script cannot use:

```
test -b foo -o -c foo -o -d foo -o -f foo -o -p foo
```

to find out if **foo** is an existing file.) On historical BSD systems, the existence of a file could be determined by:

```
test -f foo -o -d foo
```

but there was no easy way to determine that an existing file was a regular file. An early proposal used the KornShell **-a** primary (with the same meaning), but this was changed to **-e** because there were concerns about the high probability of humans confusing the **-a** primary with the **-a** binary operator.

The **-a** and **-o** binary operators and the grouping parentheses were omitted from this volume of IEEE Std. 1003.1-200x due to a difference between historical implementations of the *test* utility in the precedence of the binary primaries **'='** and **"!="** compared to the unary primaries **-b**, **-c**, **-d**, **-f**, **-g**, **-n**, **-p**, **-r**, **-s**, **-t**, **-u**, **-w**, **-x**, and **-z**. On BSD, Version 7, PWB, and 32V systems, the unary primaries have higher precedence than the binary operators; on System III and System V implementations, the binary operators **'='** and **"!="** have higher precedence. The change was apparently made for System III so that the construct:

```
test "$1" = "$2"
```

could be made to work even if \$1 started with **'-'**. This change was considered a mistake because:

- It is not a complete solution; if \$1 expands to **'( ' or '!''**, it still does not work.
- It makes it impossible to use the unary primaries **-n** and **-z** to test for a null string if there is any chance that the string will expand to **'='**.
- More importantly, there was the well known option of specifying:

```
test X"$1" = X"$2"
```

that always worked.

Unfortunately, when the **'='** and **"!="** binary primaries were given precedence over the unary primaries, there was no solution provided for scripts that wanted to reliably specify something like:

```
test -n "$1"
```

because if \$1 expands to **'='**, it gives a syntax error.

There was some discussion of forbidding the System V behavior and requiring the more logical precedence that originated in its predecessors and that remains in BSD-based systems. However, there are simply too many historical applications that would break if System V were required to make this change; this number dwarfed the number of scripts using combination logic that would then no longer be strictly portable.

This volume of IEEE Std. 1003.1-200x requires that if *test* is called with one, two, three, or four operands it correctly interprets the expression even if there is an alternate syntax tree that could lead to a syntax error. It eliminates the requirement that many string comparisons be protected with leading characters, such as:

```
test X"$1" = X"$2"
```

and allows the single-argument *string* form to be used with all possible inputs.

36454 The following examples show some of the changes that are required to be made to make  
 36455 historical BSD and System V-based implementations of *test* conform to this volume of  
 36456 IEEE Std. 1003.1-200x:

```
36457 test -d = POSIX.2 True if there is a directory named =.

36458 BSD True if there is a directory named =.

36459 System V Syntax error; = needs two operands.
```

```
36460 test -d = -f POSIX.2 False

36461 BSD Syntax error; it expects -a or -o after -d =.

36462 System V False
```

36463 Implementations are prohibited from extending *test* with options because it would make the *test*  
 36464 *string* case ambiguous for inputs that might match an extended option. Implementations can  
 36465 add primaries and operators, as indicated.

36466 The following options were not included in this volume of IEEE Std. 1003.1-200x, although they  
 36467 are provided by some historical implementations, since these facilities and concepts are not  
 36468 supported by the System Interfaces volume of IEEE Std. 1003.1-200x nor defined in this volume  
 36469 of IEEE Std. 1003.1-200x. These operands should not be used by new implementations for other  
 36470 purposes.

```
36471 -h file True if file exists and is a symbolic link.

36472 -k file True if file exists and its sticky bit is set.

36473 -L file True if file is a symbolic link.

36474 -C file True if file is a contiguous file.

36475 -S file True if file is a socket.

36476 -V file True if file is a version file.
```

36477 The following option was not included because it was undocumented in most implementations,  
 36478 has been removed from some implementations (including System V), and the functionality is  
 36479 provided by the shell (see Section 2.6.2 on page 51).

```
36480 -l string The length of the string string.
```

36481 The **-b**, **-c**, **-g**, **-p**, **-u**, and **-x** operands are derived from the SVID; historical BSD does not  
 36482 provide them. The **-k** operand is derived from System V; historical BSD does not provide it.

36483 On historical BSD systems, *test -w directory* always returned false because *test* tried to open the  
 36484 directory for writing, which always fails.

36485 Some additional primaries newly invented or from the KornShell appeared in an early proposal  
 36486 as part of the conditional command ([[]]): *s1 > s2*, *s1 < s2*, *str = pattern*, *str != pattern*, *f1 -nt f2*, *f1*  
 36487 *-ot f2*, and *f1 -ef f2*. They were not carried forward into the *test* utility when the conditional  
 36488 command was removed from the shell because they have not been included in the *test* utility  
 36489 built into historical implementations of the *sh* utility.

36490 The **-t file\_descriptor** primary is shown with a mandatory argument because the grammar is  
 36491 ambiguous if it can be omitted. Historical implementations have allowed it to be omitted,  
 36492 providing a default of 1.

#### 36493 FUTURE DIRECTIONS

36494 None.

|       |                                                                                                         |  |
|-------|---------------------------------------------------------------------------------------------------------|--|
| 36495 | <b>SEE ALSO</b>                                                                                         |  |
| 36496 | <i>find</i>                                                                                             |  |
| 36497 | <b>CHANGE HISTORY</b>                                                                                   |  |
| 36498 | First released in Issue 2.                                                                              |  |
| 36499 | <b>Issue 4</b>                                                                                          |  |
| 36500 | Aligned with the ISO/IEC 9945-2: 1993 standard.                                                         |  |
| 36501 | <b>Issue 5</b>                                                                                          |  |
| 36502 | FUTURE DIRECTIONS section added.                                                                        |  |
| 36503 | <b>Issue 6</b>                                                                                          |  |
| 36504 | The <b>-h</b> operand is added for symbolic links, and access permission requirements are clarified for |  |
| 36505 | the <b>-r</b> , <b>-w</b> , and <b>-x</b> operands to align with the IEEE P1003.2b draft standard.      |  |
| 36506 | The normative text is reworded to avoid use of the term “must” for application requirements.            |  |

36507 **NAME**

36508           time — time a simple command

36509 **SYNOPSIS**36510 UP       time [-p] *utility* [*argument...*]

36511

36512 **DESCRIPTION**

36513       The *time* utility shall invoke the utility named by the *utility* operand with arguments supplied as  
 36514       the *argument* operands and write a message to standard error that lists timing statistics for the  
 36515       utility. The message shall include the following information:

- 36516           • The elapsed (real) time between invocation of *utility* and its termination.
- 36517           • The User CPU time, equivalent to the sum of the *tms\_utime* and *tms\_cutime* fields returned by  
 36518           the *times()* function defined in the System Interfaces volume of IEEE Std. 1003.1-200x for the  
 36519           process in which *utility* is executed.
- 36520           • The System CPU time, equivalent to the sum of the *tms\_stime* and *tms\_cstime* fields returned  
 36521           by the *times()* function for the process in which *utility* is executed.

36522       The precision of the timing shall be no less than the granularity defined for the size of the clock  
 36523       tick unit on the system, but the results shall be reported in terms of standard time units (for  
 36524       example, 0.02 seconds, 00:00:00.02, 1m33.75s, 365.21 seconds), not numbers of clock ticks.

36525       When *time* is used as part of a pipeline, the times reported are unspecified, except when it is the  
 36526       sole command within a grouping command (see Section 2.9.4.1 on page 75) in that pipeline. For  
 36527       example, the commands on the left are unspecified; those on the right report on utilities **a** and **c**,  
 36528       respectively:

```
36529 time a | b | c { time a } | b | c
36530 a | b | time c a | b | (time c)
```

36531 **OPTIONS**

36532       The *time* utility shall conform to the System Interface Definitions volume of |  
 36533       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines. |

36534       The following option shall be supported:

36535       **-p**           Write the timing output to standard error in the format shown in the STDERR  
 36536       section.

36537 **OPERANDS**

36538       The following operands shall be supported:

36539       *utility*       The name of a utility that is to be invoked. If the *utility* operand names any of the  
 36540       special built-in utilities in Section 2.14 on page 96, the results are undefined.

36541       *argument*     Any string to be supplied as an argument when invoking the utility named by the  
 36542       *utility* operand.

36543 **STDIN**

36544       Not used.

36545 **INPUT FILES**

36546       None.

36547 **ENVIRONMENT VARIABLES**

36548 The following environment variables shall affect the execution of *time*:

36549 **LANG** Provide a default value for the internationalization variables that are unset or null.  
 36550 If *LANG* is unset or null, the corresponding value from the implementation-  
 36551 dependent default locale shall be used. If any of the internationalization variables  
 36552 contains an invalid setting, the utility shall behave as if none of the variables had  
 36553 been defined.

36554 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 36555 internationalization variables.

36556 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 36557 characters (for example, single-byte as opposed to multi-byte characters in  
 36558 arguments).

36559 **LC\_MESSAGES**  
 36560 Determine the locale that should be used to affect the format and contents of  
 36561 diagnostic and informative messages written to standard error.

36562 **LC\_NUMERIC**  
 36563 Determine the locale for numeric formatting.

36564 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

36565 **PATH** Determine the search path that shall be used to locate the utility to be invoked; see  
 36566 the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 8,  
 36567 Environment Variables.

36568 **ASYNCHRONOUS EVENTS**

36569 Default.

36570 **STDOUT**

36571 Not used.

36572 **STDERR**

36573 The standard error shall be used to write the timing statistics. If **-p** is specified, the following  
 36574 format shall be used in the POSIX locale:

36575 "real %f\nuser %f\nsys %f\n", <real seconds>, <user seconds>,  
 36576 <system seconds>

36577 where each floating-point number shall be expressed in seconds. The precision used may be less  
 36578 than the default six digits of %f, but shall be sufficiently precise to accommodate the size of the  
 36579 clock tick on the system (for example, if there were 60 clock ticks per second, at least two digits  
 36580 shall follow the radix character). The number of digits following the radix character shall be no  
 36581 less than one, even if this always results in a trailing zero. The implementation may append  
 36582 white space and additional information following the format shown here.

36583 **OUTPUT FILES**

36584 None.

36585 **EXTENDED DESCRIPTION**

36586 None.

36587 **EXIT STATUS**

36588 If the *utility* utility is invoked, the exit status of *time* shall be the exit status of *utility*; otherwise,  
 36589 the *time* utility shall exit with one of the following values:



- 36590 1-125 An error occurred in the *time* utility.
- 36591 126 The utility specified by *utility* was found but could not be invoked.
- 36592 127 The utility specified by *utility* could not be found.

### 36593 CONSEQUENCES OF ERRORS

36594 Default.

### 36595 APPLICATION USAGE

36596 The *command*, *env*, *nice*, *nohup*, *time*, and *xargs* utilities have been specified to use exit code 127 if  
 36597 an error occurs so that applications can distinguish “failure to find a utility” from “invoked  
 36598 utility exited with an error indication”. The value 127 was chosen because it is not commonly  
 36599 used for other meanings; most utilities use small values for “normal error conditions” and the  
 36600 values above 128 can be confused with termination due to receipt of a signal. The value 126 was  
 36601 chosen in a similar manner to indicate that the utility could be found, but not invoked. Some  
 36602 scripts produce meaningful error messages differentiating the 126 and 127 cases. The distinction  
 36603 between exit codes 126 and 127 is based on KornShell practice that uses 127 when all attempts to  
 36604 *exec* the utility fail with [ENOENT], and uses 126 when any attempt to *exec* the utility fails for  
 36605 any other reason.

36606 Application writers should note that this utility need not be provided on systems that do not  
 36607 support the User Portability Utilities option.

### 36608 EXAMPLES

36609 It is frequently desirable to apply *time* to pipelines or lists of commands. This can be done by  
 36610 placing pipelines and command lists in a single file; this file can then be invoked as a utility, and  
 36611 the *time* applies to everything in the file.

36612 Alternatively, the following command can be used to apply *time* to a complex command:

36613 `time sh -c 'complex-command-line'`

### 36614 RATIONALE

36615 The *time* utility when originally proposed for this volume of IEEE Std. 1003.1-200x, was rejected  
 36616 because it was not useful for portable applications:

- 36617 • The underlying CPU definitions from the System Interfaces volume of IEEE Std. 1003.1-200x  
 36618 are vague, so the numeric output could not be compared accurately between systems or even  
 36619 between invocations.
- 36620 • The creation of portable benchmark programs was outside the scope this volume of  
 36621 IEEE Std. 1003.1-200x.

36622 However, *time* does fit in the scope of user portability. Human judgement can be applied to the  
 36623 analysis of the output, and it could be very useful in hands-on debugging of applications or in  
 36624 providing subjective measures of system performance. Hence it has been included in this  
 36625 volume of IEEE Std. 1003.1-200x.

36626 The default output format has been left unspecified because historical implementations differ  
 36627 greatly in their style of depicting this numeric output. The **-p** option was invented to provide  
 36628 scripts a common means of obtaining this information.

36629 In the KornShell, *time* is a shell reserved word that can be used to time an entire pipeline, rather  
 36630 than just a simple command. The POSIX definition has been worded to allow this  
 36631 implementation. Consideration was given to invalidating this approach because of the historical  
 36632 model from the C shell and System V shell. However, since the System V *time* utility historically  
 36633 has not produced accurate results in pipeline timing (because the constituent processes are not  
 36634 all owned by the same parent process, as allowed by POSIX), it did not seem worthwhile to

- 36635 break historical KornShell usage.
- 36636 The term *utility* is used, rather than *command*, to highlight the fact that shell compound  
36637 commands, pipelines, special built-ins, and so on, cannot be used directly. However, *utility*  
36638 includes user application programs and shell scripts, not just the standard utilities.
- 36639 **FUTURE DIRECTIONS**
- 36640 None.
- 36641 **SEE ALSO**
- 36642 *sh*, the System Interfaces volume of IEEE Std. 1003.1-200x, *times()*
- 36643 **CHANGE HISTORY**
- 36644 First released in Issue 2.
- 36645 **Issue 4**
- 36646 Aligned with the ISO/IEC 9945-2: 1993 standard.
- 36647 **Issue 6**
- 36648 This utility is now marked as part of the User Portability Utilities option.

36649 **NAME**

36650 touch — change file access and modification times

36651 **SYNOPSIS**36652 touch [-acm][ -r *ref\_file*| -t *time*] *file*...36653 **DESCRIPTION**

36654 The *touch* utility shall change the modification times, access times, or both of files. The  
 36655 modification time shall be equivalent to the value of the *st\_mtime* member of the **stat** structure  
 36656 for a file, as described in the System Interfaces volume of IEEE Std. 1003.1-200x; the access time  
 36657 shall be equivalent to the value of *st\_atime*.

36658 The time used can be specified by the -t *time* option-argument, the corresponding time fields of  
 36659 the file referenced by the -r *ref\_file* option-argument, or the *date\_time* operand, as specified in the  
 36660 following sections. If none of these are specified, *touch* shall use the current time (the value  
 36661 returned by the equivalent of the *time()* function defined in the System Interfaces volume of  
 36662 IEEE Std. 1003.1-200x).

36663 For each *file* operand, *touch* shall perform actions equivalent to the following functions defined  
 36664 in the System Interfaces volume of IEEE Std. 1003.1-200x:

- 36665 1. If *file* does not exist, a *creat()* function call is made with the *file* operand used as the *path*  
 36666 argument and the value of the bitwise-inclusive OR of S\_IRUSR, S\_IWUSR, S\_IRGRP,  
 36667 S\_IWGRP, S\_IROTH, and S\_IWOTH used as the *mode* argument.
- 36668 2. The *utime()* function is called with the following arguments:
  - 36669 a. The *file* operand is used as the *path* argument.
  - 36670 b. The **utimbuf** structure members *actime* and *modtime* are determined as described in  
 36671 the OPTIONS section.

36672 **OPTIONS**

36673 The *touch* utility shall conform to the System Interface Definitions volume of  
 36674 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

36675 The following options shall be supported:

- |                |                           |                                                                                                                             |
|----------------|---------------------------|-----------------------------------------------------------------------------------------------------------------------------|
| 36676<br>36677 | <b>-a</b>                 | Change the access time of <i>file</i> . Do not change the modification time unless <b>-m</b> is also specified.             |
| 36678<br>36679 | <b>-c</b>                 | Do not create a specified <i>file</i> if it does not exist. Do not write any diagnostic messages concerning this condition. |
| 36680<br>36681 | <b>-m</b>                 | Change the modification time of <i>file</i> . Do not change the access time unless <b>-a</b> is also specified.             |
| 36682<br>36683 | <b>-r <i>ref_file</i></b> | Use the corresponding time of the file named by the path name <i>ref_file</i> instead of the current time.                  |
| 36684<br>36685 | <b>-t <i>time</i></b>     | Use the specified <i>time</i> instead of the current time. The option-argument shall be a decimal number of the form:       |
| 36686          |                           | [ [ <i>CC</i> ] <i>YY</i> ] <i>MMDDhhmm</i> [ . <i>SS</i> ]                                                                 |
| 36687          |                           | where each two digits represents the following:                                                                             |
| 36688          | <i>MM</i>                 | The month of the year [01-12].                                                                                              |
| 36689          | <i>DD</i>                 | The day of the month [01-31].                                                                                               |

36690 *hh* The hour of the day [00-23].

36691 *mm* The minute of the hour [00-59].

36692 *CC* The first two digits of the year (the century).

36693 *YY* The second two digits of the year.

36694 *SS* The second of the minute [00-61].

36695 Both *CC* and *YY* shall be optional. If neither is given, the current year shall be  
36696 assumed. If *YY* is specified, but *CC* is not, *CC* shall be derived as follows:

|       |                  |                    |
|-------|------------------|--------------------|
| 36697 | <b>If YY is:</b> | <b>CC becomes:</b> |
| 36698 | 69-99            | 19                 |
| 36699 | 00-68            | 20                 |

36700 The resulting time shall be affected by the value of the *TZ* environment variable. If  
36701 the resulting time value precedes the Epoch, *touch* shall exit immediately with an  
36702 error status. The range of valid times past the Epoch is implementation-dependent,  
36703 MAN but it shall extend to at least the time 0 hours, 0 minutes, 0 seconds, January 1,  
36704 2038, Coordinated Universal Time. Some systems may not be able to represent  
36705 dates beyond the January 18, 2038, because they use **signed int** as a time holder.

36706 The range for *SS* is (00-61) rather than (00-59) because of leap seconds. If *SS* is 60 or  
36707 61, and the resulting time, as affected by the *TZ* environment variable, does not  
36708 refer to a leap second, the resulting time shall be one or two seconds after a time  
36709 where *SS* is 59. If *SS* is not given a value, it is assumed to be zero.

36710 If neither the **-a** nor **-m** options were specified, *touch* shall behave as if both the **-a** and **-m**  
36711 options were specified.

## 36712 OPERANDS

36713 The following operands shall be supported:

36714 *file* A path name of a file whose times shall be modified. |

## 36715 STDIN

36716 Not used.

## 36717 INPUT FILES

36718 None.

## 36719 ENVIRONMENT VARIABLES

36720 The following environment variables shall affect the execution of *touch*:

36721 *LANG* Provide a default value for the internationalization variables that are unset or null.  
36722 If *LANG* is unset or null, the corresponding value from the implementation-  
36723 dependent default locale shall be used. If any of the internationalization variables  
36724 contains an invalid setting, the utility shall behave as if none of the variables had  
36725 been defined.

36726 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
36727 internationalization variables.

36728 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
36729 characters (for example, single-byte as opposed to multi-byte characters in  
36730 arguments).

36731 *LC\_MESSAGES*

36732 Determine the locale that should be used to affect the format and contents of

36733 diagnostic messages written to standard error.

36734 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

36735 **TZ** Determine the timezone to be used for interpreting the *time* option-argument.

36736 **ASYNCHRONOUS EVENTS**

36737 Default.

36738 **STDOUT**

36739 Not used.

36740 **STDERR**

36741 Used only for diagnostic messages.

36742 **OUTPUT FILES**

36743 None.

36744 **EXTENDED DESCRIPTION**

36745 None.

36746 **EXIT STATUS**

36747 The following exit values shall be returned:

36748 0 The utility executed successfully and all requested changes were made.

36749 >0 An error occurred.

36750 **CONSEQUENCES OF ERRORS**

36751 Default.

36752 **APPLICATION USAGE**

36753 The interpretation of time is taken to be *seconds since the Epoch* (see the System Interface  
 36754 Definitions volume of IEEE Std. 1003.1-200x, Section 3.342, Seconds Since the Epoch). It should  
 36755 be noted that implementations conforming to the System Interfaces volume of  
 36756 IEEE Std. 1003.1-200x do not take leap seconds into account when computing seconds since the  
 36757 Epoch. When SS=60 is used, the resulting time always refers to 1 plus *seconds since the Epoch* for a  
 36758 time when SS=59.

36759 Although the **-t time** option-argument and the obsolescent *date\_time* operand specify values in  
 36760 1969, the access time and modification time fields are defined in terms of seconds since the  
 36761 Epoch (midnight on 1 January 1970 UTC). Therefore, depending on the value of *TZ* when *touch* is  
 36762 run, there is never more than a few valid hours in 1969 and there need not be any valid times in  
 36763 1969.

36764 One ambiguous situation occurs if **-t time** is not specified, **-r ref\_file** is not specified, and the first  
 36765 operand is an eight or ten-digit decimal number. A portable script can avoid this problem by  
 36766 using:

36767 touch -- file

36768 or:

36769 touch ./file

36770 in this case.

36771 **EXAMPLES**

36772 None.

36773 **RATIONALE**

36774 The functionality of *touch* is described almost entirely through references to functions in the  
 36775 System Interfaces volume of IEEE Std. 1003.1-200x. In this way, there is no duplication of effort  
 36776 required for describing such side effects as the relationship of user IDs to the user database,  
 36777 permissions, and so on.

36778 There are some significant differences between the *touch* utility in this volume of  
 36779 IEEE Std. 1003.1-200x and those in System V and BSD systems. They are upward-compatible for  
 36780 historical applications from both implementations:

36781 1. In System V, an ambiguity exists when a path name that is a decimal number leads the  
 36782 operands; it is treated as a time value. In BSD, no *time* value is allowed; files may only be  
 36783 *touched* to the current time. The *-t time* construct solves these problems for future portable  
 36784 applications (note that the *-t* option is not historical practice).

36785 2. The inclusion of the century digits, *CC*, is also new. Note that a ten-digit *time* value is  
 36786 treated as if *YY*, and not *CC*, were specified. The caveat about the range of dates following  
 36787 the Epoch was included as recognition that some implementations are not able to  
 36788 represent dates beyond 18 January 2038 because they use **signed int** as a time holder.

36789 The *-r* option was added because several comments requested this capability. This option was  
 36790 named *-f* in an early proposal, but was changed because the *-f* option is used in the BSD version  
 36791 of *touch* with a different meaning.

36792 At least one historical implementation of *touch* incremented the exit code if *-c* was specified and  
 36793 the file did not exist. This volume of IEEE Std. 1003.1-200x requires exit status zero if no errors  
 36794 occur.

36795 **FUTURE DIRECTIONS**

36796 Applications should use the *-r* or *-t* options.

36797 **SEE ALSO**

36798 *date*, the System Interfaces volume of IEEE Std. 1003.1-200x, *creat()*, *time()*, <sys/stat.h>

36799 **CHANGE HISTORY**

36800 First released in Issue 2.

36801 **Issue 4**

36802 Aligned with the ISO/IEC 9945-2:1993 standard.

36803 **Issue 6**

36804 The obsolescent *date\_time* operand is removed.

36805 The Open Group corrigenda item U027/1 has been applied. This extends the range of valid time  
 36806 past the Epoch to at least the time 0 hours, 0 minutes, 0 seconds, January 1, 2038, Coordinated  
 36807 Universal Time. This is a new requirement on POSIX implementations.

36808 **Notes to Reviewers**

36809 *This section with side shading will not appear in the final copy. - Ed.*

36810 Should leap seconds be 00-61? c9x infers that it is only 00-60, and astronomers confirm that  
 36811 double leap seconds do not occur.

36812 **NAME**

36813           tput — change terminal characteristics

36814 **SYNOPSIS**36815 UP       tput [-T *type*] *operand...*

36816

36817 **DESCRIPTION**

36818       The *tput* utility shall display terminal-dependent information. The manner in which this  
 36819       information is retrieved is unspecified. The information displayed shall clear the terminal screen,  
 36820       initialize the user's terminal, or reset the user's terminal, depending on the operand given. The  
 36821       exact consequences of displaying this information are unspecified.

36822 **OPTIONS**

36823       The *tput* utility shall conform to the System Interface Definitions volume of  
 36824       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

36825       The following option shall be supported:

36826       -T *type*     Indicate the type of terminal. If this option is not supplied and the *TERM* variable  
 36827       is unset or null, an unspecified default terminal type shall be used. The setting of  
 36828       *type* shall take precedence over the value in *TERM*.

36829 **OPERANDS**

36830       The following strings shall be supported as operands by the implementation in the POSIX locale:

36831       **clear**       Display the clear-screen sequence.

36832       **init**        Display the sequence that initializes the user's terminal in an implementation-  
 36833       dependent manner.

36834       **reset**       Display the sequence that resets the user's terminal in an implementation-  
 36835       dependent manner.

36836       If a terminal does not support any of the operations described by these operands, this shall not  
 36837       be considered an error condition.

36838 **STDIN**

36839       Not used.

36840 **INPUT FILES**

36841       None.

36842 **ENVIRONMENT VARIABLES**

36843       The following environment variables shall affect the execution of *tput*:

36844       **LANG**        Provide a default value for the internationalization variables that are unset or null.  
 36845       If *LANG* is unset or null, the corresponding value from the implementation-  
 36846       dependent default locale shall be used. If any of the internationalization variables  
 36847       contains an invalid setting, the utility shall behave as if none of the variables had  
 36848       been defined.

36849       **LC\_ALL**       If set to a non-empty string value, override the values of all the other  
 36850       internationalization variables.

36851       **LC\_CTYPE**    Determine the locale for the interpretation of sequences of bytes of text data as  
 36852       characters (for example, single-byte as opposed to multi-byte characters in  
 36853       arguments).

36854       **LC\_MESSAGES**

36855       Determine the locale that should be used to affect the format and contents of

36856 diagnostic messages written to standard error.

36857 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

36858 **TERM** Determine the terminal type. If this variable is unset or null, and if the **-T** option is  
 36859 not specified, an unspecified default terminal type shall be used.

36860 **ASYNCHRONOUS EVENTS**

36861 Default.

36862 **STDOUT**

36863 If standard output is a terminal device, it may be used for writing the appropriate sequence to  
 36864 clear the screen or reset or initialize the terminal. If standard output is not a terminal device,  
 36865 undefined results occur.

36866 **STDERR**

36867 Used only for diagnostic messages.

36868 **OUTPUT FILES**

36869 None.

36870 **EXTENDED DESCRIPTION**

36871 None.

36872 **EXIT STATUS**

36873 The following exit values shall be returned:

36874 0 The requested string was written successfully.

36875 1 Unspecified.

36876 2 Usage error.

36877 3 No information is available about the specified terminal type.

36878 4 The specified operand is invalid.

36879 >4 An error occurred.

36880 **CONSEQUENCES OF ERRORS**

36881 If one of the operands is not available for the terminal, *tput* continues processing the remaining  
 36882 operands.

36883 **APPLICATION USAGE**

36884 The difference between resetting and initializing a terminal is left unspecified, as they vary  
 36885 greatly based on hardware types. In general, resetting is a more severe action.

36886 Some terminals use control characters to perform the stated functions, and on such terminals it  
 36887 might make sense to use *tput* to store the initialization strings in a file or environment variable  
 36888 for later use. However, because other terminals might rely on system calls to do this work, the  
 36889 standard output cannot be used in a portable manner, such as the following non-portable  
 36890 constructs:

36891 ClearVar='tput clear'  
 36892 tput reset | mailx -s "Wake Up" ddg

36893 Application writers should note that this utility need not be provided on systems that do not  
 36894 support the User Portability Utilities option.



36895 **EXAMPLES**

36896           1. Initialize the terminal according to the type of terminal in the environmental variable  
36897           *TERM*. This command can be included in a **.profile** file.

36898           tput init

36899           2. Reset a 450 terminal.

36900           tput -T 450 reset

36901 **RATIONALE**

36902           The list of operands was reduced to a minimum for the following reasons:

36903           • The only features chosen were those that were likely to be used by human users interacting  
36904           with a terminal.

36905           • Specifying the full *terminfo* set was not considered desirable, but the standard developers did  
36906           not want to select among operands.

36907           • This volume of IEEE Std. 1003.1-200x does not attempt to provide applications with  
36908           sophisticated terminal handling capabilities, as that falls outside of its assigned scope and  
36909           intersects with the responsibilities of other standards bodies.

36910           The difference between resetting and initializing a terminal is left unspecified as this varies  
36911           greatly based on hardware types. In general, resetting is a more severe action.

36912           The exit status of 1 is historically reserved for finding out if a Boolean operand is not set.  
36913           Although the operands were reduced to a minimum, the exit status of 1 should still be reserved  
36914           for the Boolean operands, for those sites that wish to support them.

36915 **FUTURE DIRECTIONS**

36916           None.

36917 **SEE ALSO**

36918           *stty*, *tabs*

36919 **CHANGE HISTORY**

36920           First released in Issue 4.

36921 **Issue 6**

36922           This utility is now marked as part of the User Portability Utilities option.

36923 **NAME**

36924 tr — translate characters

36925 **SYNOPSIS**36926 tr [-c | -C][-s] *string1 string2* |36927 tr -s [-c | -C] *string1* |36928 tr -d [-c | -C] *string1* |36929 tr -ds [-c | -C] *string1 string2* |36930 **DESCRIPTION**

36931 The *tr* utility shall copy the standard input to the standard output with substitution or deletion  
 36932 of selected characters. The options specified and the *string1* and *string2* operands shall control  
 36933 translations that occur while copying characters and single-character collating elements.

36934 **OPTIONS**

36935 The *tr* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 36936 Section 12.2, Utility Syntax Guidelines. |

36937 The following options shall be supported:

36938 **-c** Complement the set of values specified by *string1*. See the EXTENDED  
 36939 DESCRIPTION section. |

36940 **-C** Complement the set of characters specified by *string1*. See the EXTENDED  
 36941 DESCRIPTION section. |

36942 **-d** Delete all occurrences of input characters that are specified by *string1*.

36943 **-s** Replace instances of repeated characters with a single character, as described in the  
 36944 EXTENDED DESCRIPTION section.

36945 **OPERANDS**

36946 The following operands shall be supported:

36947 *string1, string2*

36948 Translation control strings. Each string shall represent a set of characters to be  
 36949 converted into an array of characters used for the translation. For a detailed  
 36950 description of how the strings are interpreted, see the EXTENDED DESCRIPTION  
 36951 section.

36952 **STDIN**

36953 The standard input can be any type of file.

36954 **INPUT FILES**

36955 None.

36956 **ENVIRONMENT VARIABLES**36957 The following environment variables shall affect the execution of *tr*:

36958 **LANG** Provide a default value for the internationalization variables that are unset or null.  
 36959 If *LANG* is unset or null, the corresponding value from the implementation-  
 36960 dependent default locale shall be used. If any of the internationalization variables  
 36961 contains an invalid setting, the utility shall behave as if none of the variables had  
 36962 been defined.

36963 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 36964 internationalization variables.

|       |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------|-----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 36965 | <b>LC_COLLATE</b>           |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 36966 |                             | Determine the locale for the behavior of range expressions and equivalence classes.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 36967 | <b>LC_CTYPE</b>             | Determine the locale for the interpretation of sequences of bytes of text data as characters (for example, single-byte as opposed to multi-byte characters in arguments) and the behavior of character classes.                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 36968 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 36969 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 36970 | <b>LC_MESSAGES</b>          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 36971 |                             | Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 36972 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 36973 | XSI <b>NLS_PATH</b>         | Determine the location of message catalogs for the processing of <b>LC_MESSAGES</b> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 36974 | <b>ASYNCHRONOUS EVENTS</b>  |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 36975 |                             | Default.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 36976 | <b>STDOUT</b>               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 36977 |                             | The <i>tr</i> output shall be identical to the input, with the exception of the specified transformations.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 36978 | <b>STDERR</b>               |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 36979 |                             | Used only for diagnostic messages.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 36980 | <b>OUTPUT FILES</b>         |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 36981 |                             | None.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 36982 | <b>EXTENDED DESCRIPTION</b> |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 36983 |                             | The operands <i>string1</i> and <i>string2</i> (if specified) define two arrays of characters. The constructs in the following list can be used to specify characters or single-character collating elements. If any of the constructs result in multi-character collating elements, <i>tr</i> shall exclude, without a diagnostic, those multi-character elements from the resulting array.                                                                                                                                                                                                                                                              |
| 36984 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 36985 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 36986 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 36987 | <i>character</i>            | Any character not described by one of the conventions below represents itself.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 36988 | <i>\octal</i>               | Octal sequences can be used to represent characters with specific coded values. An octal sequence shall consist of a backslash followed by the longest sequence of one, two, or three-octal-digit characters (01234567). The sequence shall cause the value whose encoding is represented by the one, two, or three-digit octal integer to be placed into the array. If the size of a byte on the system is greater than nine bits, the valid escape sequence used to represent a byte is implementation-dependent. Multi-byte characters require multiple, concatenated escape sequences of this type, including the leading ' <i>\</i> ' for each byte. |
| 36989 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 36990 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 36991 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 36992 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 36993 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 36994 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 36995 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 36996 | <i>\character</i>           | The backslash-escape sequences in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Table 5-1, Escape Sequences and Associated Actions (' <i>\</i> <i>\</i> ', ' <i>\</i> <i>a</i> ', ' <i>\</i> <i>b</i> ', ' <i>\</i> <i>f</i> ', ' <i>\</i> <i>n</i> ', ' <i>\</i> <i>r</i> ', ' <i>\</i> <i>t</i> ', ' <i>\</i> <i>v</i> ') shall be supported. The results of using any other character, other than an octal digit, following the backslash are unspecified.                                                                                                                                                                         |
| 36997 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 36998 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 36999 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 37000 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 37001 | <i>c-c</i>                  | Represents the range of collating elements between the range endpoints (as long as neither endpoint is an octal sequence of the form <i>\octal</i> ), inclusive, as defined by the current setting of the <b>LC_COLLATE</b> locale category. The application shall ensure that the starting endpoint precedes the second endpoint in the current collation order. The characters or collating elements in the range shall be placed in the array in ascending collation sequence.                                                                                                                                                                         |
| 37002 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 37003 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 37004 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 37005 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 37006 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 37007 |                             | If either or both of the range endpoints are octal sequences of the form <i>\octal</i> , this shall represent the range of specific coded values between the two range endpoints, inclusive.                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 37008 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 37009 |                             |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

|           |                         |                                                                                                                         |
|-----------|-------------------------|-------------------------------------------------------------------------------------------------------------------------|
| 37010     | [:class:]               | Represents all characters belonging to the defined character class, as defined by the                                   |
| 37011     |                         | current setting of the <i>LC_CTYPE</i> locale category. The following character class                                   |
| 37012     |                         | names shall be accepted when specified in <i>string1</i> :                                                              |
| 37013     |                         | <b>alnum</b> <b>blank</b> <b>digit</b> <b>lower</b> <b>punct</b> <b>upper</b>                                           |
| 37014     |                         | <b>alpha</b> <b>cntrl</b> <b>graph</b> <b>print</b> <b>space</b> <b>xdigit</b>                                          |
| 37015 XSI |                         | In addition, character class expressions of the form [: <i>name</i> :] shall be recognized in                           |
| 37016     |                         | those locales where the <i>name</i> keyword has been given a <b>charclass</b> definition in the                         |
| 37017     |                         | <i>LC_CTYPE</i> category.                                                                                               |
| 37018     |                         | When both the <b>-d</b> and <b>-s</b> options are specified, any of the character class names                           |
| 37019     |                         | shall be accepted in <i>string2</i> . Otherwise, only character class names <b>lower</b> or <b>upper</b>                |
| 37020     |                         | are valid in <i>string2</i> and then only if the corresponding character class ( <b>upper</b> and                       |
| 37021     |                         | <b>lower</b> , respectively) is specified in the same relative position in <i>string1</i> . Such a                      |
| 37022     |                         | specification shall be interpreted as a request for case conversion. When [: <i>lower</i> :]                            |
| 37023     |                         | appears in <i>string1</i> and [: <i>upper</i> :] appears in <i>string2</i> , the arrays shall contain the               |
| 37024     |                         | characters from the <b>toupper</b> mapping in the <i>LC_CTYPE</i> category of the current                               |
| 37025     |                         | locale. When [: <i>upper</i> :] appears in <i>string1</i> and [: <i>lower</i> :] appears in <i>string2</i> , the arrays |
| 37026     |                         | shall contain the characters from the <b>tolower</b> mapping in the <i>LC_CTYPE</i> category                            |
| 37027     |                         | of the current locale. The first character from each mapping pair shall be in the                                       |
| 37028     |                         | array for <i>string1</i> and the second character from each mapping pair shall be in the                                |
| 37029     |                         | array for <i>string2</i> in the same relative position.                                                                 |
| 37030     |                         | Except for case conversion, the characters specified by a character class expression                                    |
| 37031     |                         | shall be placed in the array in an unspecified order.                                                                   |
| 37032     |                         | If the name specified for <i>class</i> does not define a valid character class in the current                           |
| 37033     |                         | locale, the behavior is undefined.                                                                                      |
| 37034     | [= <i>equiv</i> =]      | Represents all characters or collating elements belonging to the same equivalence                                       |
| 37035     |                         | class as <i>equiv</i> , as defined by the current setting of the <i>LC_COLLATE</i> locale                               |
| 37036     |                         | category. An equivalence class expression shall be allowed only in <i>string1</i> , or in                               |
| 37037     |                         | <i>string2</i> when it is being used by the combined <b>-d</b> and <b>-s</b> options. The characters                    |
| 37038     |                         | belonging to the equivalence class shall be placed in the array in an unspecified                                       |
| 37039     |                         | order.                                                                                                                  |
| 37040     | [ <i>x</i> * <i>n</i> ] | Represents <i>n</i> repeated occurrences of the character <i>x</i> . Because this expression is                         |
| 37041     |                         | used to map multiple characters to one, it is only valid when it occurs in <i>string2</i> . If                          |
| 37042     |                         | <i>n</i> is omitted or is zero, it shall be interpreted as large enough to extend the <i>string2</i> -                  |
| 37043     |                         | based sequence to the length of the <i>string1</i> -based sequence. If <i>n</i> has a leading zero,                     |
| 37044     |                         | it shall be interpreted as an octal value. Otherwise, it shall be interpreted as a                                      |
| 37045     |                         | decimal value.                                                                                                          |
| 37046     |                         | When the <b>-d</b> option is not specified:                                                                             |
| 37047     |                         | • Each input character found in the array specified by <i>string1</i> shall be replaced by the character                |
| 37048     |                         | in the same relative position in the array specified by <i>string2</i> . When the array specified by                    |
| 37049     |                         | <i>string2</i> is shorter than the one specified by <i>string1</i> , the results are unspecified.                       |
| 37050     |                         | • If the <b>-C</b> option is specified, the complements of the characters specified by <i>string1</i> (the set of       |
| 37051     |                         | all characters in the current character set, as defined by the current setting of <i>LC_CTYPE</i> ,                     |
| 37052     |                         | except for those actually specified in the <i>string1</i> operand) shall be placed in the array in                      |
| 37053     |                         | ascending collation sequence, as defined by the current setting of <i>LC_COLLATE</i> .                                  |
| 37054     |                         | • If the <b>-c</b> option is specified, the complement of the values specified by <i>string1</i> shall be placed        |
| 37055     |                         | in the array in ascending order by binary value.                                                                        |

37056 • Because the order in which characters specified by character class expressions or equivalence  
 37057 class expressions is undefined, such expressions should only be used if the intent is to map  
 37058 several characters into one. An exception is case conversion, as described previously.

37059 When the **-d** option is specified:

- 37060 • Input characters found in the array specified by *string1* shall be deleted.
- 37061 • When the **-C** option is specified with **-d**, all characters except those specified by *string1* shall  
 37062 be deleted. The contents of *string2* are ignored, unless the **-s** option is also specified.
- 37063 • When the **-c** option is specified with **-d**, all values except those specified by *string1* shall be  
 37064 deleted. The contents of *string2* shall be ignored, unless the **-s** option is also specified.
- 37065 • The same string cannot be used for both the **-d** and the **-s** option; when both options are  
 37066 specified, both *string1* (used for deletion) and *string2* (used for squeezing) shall be required.

37067 When the **-s** option is specified, after any deletions or translations have taken place, repeated  
 37068 sequences of the same character shall be replaced by one occurrence of the same character, if the  
 37069 character is found in the array specified by the last operand. If the last operand contains a  
 37070 character class, such as the following example:

37071 `tr -s '[:space:]'`

37072 the last operand's array shall contain all of the characters in that character class. However, in a  
 37073 case conversion, as described previously, such as:

37074 `tr -s '[:upper:]' '[:lower:]'`

37075 the last operand's array shall contain only those characters defined as the second characters in  
 37076 each of the **toupper** or **tolower** character pairs, as appropriate.

37077 An empty string used for *string1* or *string2* produces undefined results.

#### 37078 EXIT STATUS

37079 The following exit values shall be returned:

37080     **0**   All input was processed successfully.

37081     **>0**   An error occurred.

#### 37082 CONSEQUENCES OF ERRORS

37083     Default.

#### 37084 APPLICATION USAGE

37085 If necessary, *string1* and *string2* can be quoted to avoid pattern matching by the shell.

37086 If an ordinary digit (representing itself) is to follow an octal sequence, the octal sequence must  
 37087 use the full three digits to avoid ambiguity.

37088 When *string2* is shorter than *string1*, a difference results between historical System V and BSD  
 37089 systems. A BSD system pads *string2* with the last character found in *string2*. Thus, it is possible  
 37090 to do the following:

37091 `tr 0123456789 d`

37092 which would translate all digits to the letter 'd'. Since this area is specifically unspecified in  
 37093 this volume of IEEE Std. 1003.1-200x, both the BSD and System V behaviors are allowed, but a  
 37094 portable application cannot rely on the BSD behavior. It would have to code the example in the  
 37095 following way:

37096 `tr 0123456789 '[d*]'`

37097 It should be noted that, despite similarities in appearance, the string operands used by *tr* are not  
 37098 regular expressions.

37099 Unlike some historical implementations, this definition of the *tr* utility correctly processes NUL  
 37100 characters in its input stream. NUL characters can be stripped by using:

37101 `tr -d '\000'`

### 37102 EXAMPLES

37103 1. The following example creates a list of all words in **file1** one per line in **file2**, where a word  
 37104 is taken to be a maximal string of letters.

37105 `tr -cs "[:alpha:]" "[\n*]" <file1 >file2`

37106 2. The next example translates all lowercase characters in **file1** to uppercase and writes the  
 37107 results to standard output.

37108 `tr "[:lower:]" "[:upper:]" <file1`

37109 Note that the caveat expressed in the corresponding Issue 3 example is no longer in effect.  
 37110 This case conversion is now a special case that employs the **tolower** and **toupper**  
 37111 classifications, ensuring that proper mapping is accomplished (when the locale is correctly  
 37112 defined).

37113 3. This example uses an equivalence class to identify accented variants of the base character  
 37114 'e' in **file1**, which are stripped of diacritical marks and written to **file2**.

37115 `tr "[=e]" e <file1 >file2`

### 37116 RATIONALE

37117 In some early proposals, an explicit option **-n** was added to disable the historical behavior of  
 37118 stripping NUL characters from the input. It was considered that automatically stripping NUL  
 37119 characters from the input was not correct functionality. However, the removal of **-n** in a later  
 37120 proposal does not remove the requirement that *tr* correctly process NUL characters in its input  
 37121 stream. NUL characters can be stripped by using `tr -d '\000'`.

37122 Historical implementations of *tr* differ widely in syntax and behavior. For example, the BSD  
 37123 version has not needed the bracket characters for the repetition sequence. The POSIX Shell and  
 37124 Utilities *tr* syntax is based more closely on the System V and XPG3 model while attempting to  
 37125 accommodate historical BSD implementations. In the case of the short *string2* padding, the  
 37126 decision was to unspecify the behavior and preserve System V and XPG3 scripts, which might  
 37127 find difficulty with the BSD method. The assumption was made that BSD users of *tr* have to  
 37128 make accommodations to meet the POSIX Shell and Utilities syntax. Since it is possible to use  
 37129 the repetition sequence to duplicate the desired behavior, whereas there is no simple way to  
 37130 achieve the System V method, this was the correct, if not desirable, approach.

37131 The use of octal values to specify control characters, while having historical precedents, is not  
 37132 portable. The introduction of escape sequences for control characters should provide the  
 37133 necessary portability. It is recognized that this may cause some historical scripts to break.

37134 An early proposal included support for multi-character collating elements. It was pointed out  
 37135 that, while *tr* does employ some syntactical elements from REs, the aim of *tr* is quite different;  
 37136 ranges, for example, do not have a similar meaning ("any of the chars in the range matches",  
 37137 versus "translate each character in the range to the output counterpart"). As a result, the  
 37138 previously included support for multi-character collating elements has been removed. What  
 37139 remains are ranges in current collation order (to support, for example, accented characters),  
 37140 character classes, and equivalence classes.

37141 In XPG3 the [:class:] and [=equiv=] conventions are shown with double brackets, as in RE syntax.  
 37142 However, *tr* does not implement RE principles; it just borrows part of the syntax. Consequently,  
 37143 [:class:] and [=equiv=] should be regarded as syntactical elements on a par with [x\*n], which is  
 37144 not an RE bracket expression.

37145 The standard developers will consider changes to *tr* that allow it to translate characters between  
 37146 different character encodings, or they will consider providing a new utility to accomplish this.

37147 On historical System V systems, a range expression requires enclosing square-brackets, such as:

37148 `tr '[a-z]' '[A-Z]'`

37149 However, BSD-based systems did not require the brackets, and this convention is used by POSIX  
 37150 Shell and Utilities to avoid breaking large numbers of BSD scripts:

37151 `tr a-z A-Z`

37152 The preceding System V script will continue to work because the brackets, treated as regular  
 37153 characters, are translated to themselves. However, any System V script that relied on *a-z*  
 37154 representing the three characters '-', ' and 'z' have to be rewritten as *az-*.

37155 A prior version of IEEE Std. 1003.1-200x had a *-c* option that behaved similarly to the *-C* option,  
 37156 but did not supply functionality equivalent to the *-c* option specified in IEEE Std. 1003.1-200x.  
 37157 This meant that historical practice of being able to specify *tr -d\200-\377* (which would delete  
 37158 all bytes with the top bit set) would have no effect because, in the C locale, bytes with the values  
 37159 octal 200 to octal 377 are not characters.

37160 The earlier version also said that octal sequences referred to collating elements and could be  
 37161 placed adjacent to each other to specify multi-byte characters. However, it was noted that this  
 37162 caused ambiguities because *tr* would not be able to tell whether adjacent octal sequences were  
 37163 intending to specify multi-byte characters or multiple single byte characters.  
 37164 IEEE Std. 1003.1-200x specifies that octal sequences always refer to single byte binary values.

#### 37165 **FUTURE DIRECTIONS**

37166 None.

#### 37167 **SEE ALSO**

37168 *sed*

#### 37169 **CHANGE HISTORY**

37170 First released in Issue 2.

#### 37171 **Issue 4**

37172 Aligned with the ISO/IEC 9945-2: 1993 standard.

#### 37173 **Issue 6**

37174 The *-C* operand is added, and the description of the *-c* operand is changed to align with the  
 37175 IEEE P1003.2b draft standard.

37176 The normative text is reworded to avoid use of the term “must” for application requirements.

37177 **NAME**

37178           true — return true value

37179 **SYNOPSIS**

37180           true

37181 **DESCRIPTION**

37182           The *true* utility shall return with exit code zero.

37183 **OPTIONS**

37184           None.

37185 **OPERANDS**

37186           None.

37187 **STDIN**

37188           Not used.

37189 **INPUT FILES**

37190           None.

37191 **ENVIRONMENT VARIABLES**

37192           None.

37193 **ASYNCHRONOUS EVENTS**

37194           Default.

37195 **STDOUT**

37196           Not used.

37197 **STDERR**

37198           None.

37199 **OUTPUT FILES**

37200           None.

37201 **EXTENDED DESCRIPTION**

37202           None.

37203 **EXIT STATUS**

37204           Default.

37205 **CONSEQUENCES OF ERRORS**

37206           None.

37207 **APPLICATION USAGE**

37208           This utility is typically used in shell scripts, as shown in the **EXAMPLES** section. The special  
37209           built-in utility `:` is sometimes more efficient than *true*.

37210 **EXAMPLES**

37211           This command is executed forever:

```
37212 while true
37213 do
37214 command
37215 done
```



37216 **RATIONALE**

37217 The *true* utility has been retained in this volume of IEEE Std. 1003.1-200x, even though the shell  
37218 special built-in : provides similar functionality, because *true* is widely used in historical scripts  
37219 and is less cryptic to novice script readers.

37220 **FUTURE DIRECTIONS**

37221 None.

37222 **SEE ALSO**

37223 *false*, Section 2.9 on page 67

37224 **CHANGE HISTORY**

37225 First released in Issue 2.

37226 **Issue 4**

37227 Aligned with the ISO/IEC 9945-2:1993 standard.

37228 **NAME**

37229           tsort — topological sort

37230 **SYNOPSIS**37231 XSI       tsort [*file*]

37232

37233 **DESCRIPTION**37234           The *tsort* utility shall write to standard output a totally ordered list of items consistent with a  
37235           partial ordering of items contained in the input.37236           The application shall ensure that the input consists of pairs of items (non-empty strings) |  
37237           separated by <blank>s. Pairs of different items indicate ordering. Pairs of identical items |  
37238           indicate presence, but not ordering.37239 **OPTIONS**

37240           None.

37241 **OPERANDS**

37242           The following operand shall be supported:

37243           *file*           A path name of a text file to order. If no *file* operand is given, the standard input is  
37244           used.37245 **STDIN**37246           The standard input shall be a text file that is used if no *file* operand is given. |37247 **INPUT FILES**37248           The input file named by the *file* operand is a text file.37249 **ENVIRONMENT VARIABLES**37250           The following environment variables shall affect the execution of *tsort*:37251           *LANG*           Provide a default value for the internationalization variables that are unset or null.  
37252                           If *LANG* is unset or null, the corresponding value from the implementation-  
37253                           dependent default locale shall be used. If any of the internationalization variables  
37254                           contains an invalid setting, the utility shall behave as if none of the variables had  
37255                           been defined.37256           *LC\_ALL*       If set to a non-empty string value, override the values of all the other  
37257           internationalization variables.37258           *LC\_CTYPE*   Determine the locale for the interpretation of sequences of bytes of text data as  
37259           characters (for example, single-byte as opposed to multi-byte characters in  
37260           arguments and input files).37261           *LC\_MESSAGES*                           Determine the locale that should be used to affect the format and contents of  
37262           diagnostic messages written to standard error.  
3726337264           *NLSPATH*   Determine the location of message catalogs for the processing of *LC\_MESSAGES*.37265 **ASYNCHRONOUS EVENTS**

37266           Default.

37267 **STDOUT**37268           The standard output shall be a text file consisting of the order list produced from the partially  
37269           ordered input.

**37270 STDERR**

37271           Used only for diagnostic messages.

**37272 OUTPUT FILES**

37273           None.

**37274 EXTENDED DESCRIPTION**

37275           None.

**37276 EXIT STATUS**

37277           The following exit values shall be returned:

37278           0   Successful completion.

37279           >0  An error occurred.

**37280 CONSEQUENCES OF ERRORS**

37281           Default.

**37282 APPLICATION USAGE**

37283           The *LC\_COLLATE* variable need not affect the actions of *tsort*. The output ordering is not  
37284           lexicographic, but depends on the pairs of items given as input.

**37285 EXAMPLES**

37286           The command:

37287           *tsort* <<EOF

37288           a b c c d e

37289           g g

37290           f g e f

37291           h h

37292           EOF

37293           produces the output:

37294           **a**

37295           **b**

37296           **c**

37297           **d**

37298           **e**

37299           **f**

37300           **g**

37301           **h**

**37302 RATIONALE**

37303           None.

**37304 FUTURE DIRECTIONS**

37305           None.

**37306 SEE ALSO**

37307           None.

**37308 CHANGE HISTORY**

37309           First released in Issue 2.

**37310 Issue 4**

37311           Format reorganized.

37312           Internationalized environment variable support mandated.

37313 **Issue 6**

37314 The normative text is reworded to avoid use of the term “must” for application requirements.

37315 **NAME**

37316           tty — return user's terminal name

37317 **SYNOPSIS**

37318           tty

37319 **DESCRIPTION**

37320           The *tty* utility shall write to the standard output the name of the terminal that is open as  
 37321           standard input. The name that is used shall be equivalent to the string that would be returned by  
 37322           the *ttyname()* function defined in the System Interfaces volume of IEEE Std. 1003.1-200x.

37323 **OPTIONS**

37324           The *tty* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 37325           Section 12.2, Utility Syntax Guidelines.

37326 **OPERANDS**

37327           None.

37328 **STDIN**

37329           While no input is read from standard input, standard input shall be examined to determine  
 37330           whether or not it is a terminal, and, if so, to determine the name of the terminal.

37331 **INPUT FILES**

37332           None.

37333 **ENVIRONMENT VARIABLES**37334           The following environment variables shall affect the execution of *tty*:

37335           *LANG*       Provide a default value for the internationalization variables that are unset or null.  
 37336                       If *LANG* is unset or null, the corresponding value from the implementation-  
 37337                       dependent default locale shall be used. If any of the internationalization variables  
 37338                       contains an invalid setting, the utility shall behave as if none of the variables had  
 37339                       been defined.

37340           *LC\_ALL*     If set to a non-empty string value, override the values of all the other  
 37341                       internationalization variables.

37342           *LC\_CTYPE*   Determine the locale for the interpretation of sequences of bytes of text data as  
 37343                       characters (for example, single-byte as opposed to multi-byte characters in  
 37344                       arguments).

37345           *LC\_MESSAGES*

37346                       Determine the locale that should be used to affect the format and contents of  
 37347                       diagnostic messages written to standard error and informative messages written to  
 37348                       standard output.

37349 XSI       *NLSPATH*   Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

37350 **ASYNCHRONOUS EVENTS**

37351           Default.

37352 **STDOUT**

37353           If the *-s* option is not specified and standard input is a terminal device, a path name of the  
 37354           terminal as specified by the *ttyname()* function defined in the System Interfaces volume of  
 37355           IEEE Std. 1003.1-200x shall be written in the following format:

37356           "%s\n", &lt;terminal name&gt;

37357           Otherwise, a message shall be written indicating that standard input is not connected to a  
 37358           terminal. In the POSIX locale, the *tty* utility shall use the format:

37359 "not a tty\n"

37360 **STDERR**

37361 Used only for diagnostic messages.

37362 **OUTPUT FILES**

37363 None.

37364 **EXTENDED DESCRIPTION**

37365 None.

37366 **EXIT STATUS**

37367 The following exit values shall be returned:

37368 0 Standard input is a terminal.

37369 1 Standard input is not a terminal.

37370 >1 An error occurred.

37371 **CONSEQUENCES OF ERRORS**

37372 Default.

37373 **APPLICATION USAGE**

37374 This utility checks the status of the file open as standard input against that of a system-defined  
37375 set of files. It is possible that no match can be found, or that the match found need not be the  
37376 same file as that which was opened for standard input (although they are the same device).

37377 The **-s** option is useful only if the exit code is wanted. It does not rely on the ability to form a  
37378 valid path name. Portable applications should use *test -t 0*.

37379 **EXAMPLES**

37380 None.

37381 **RATIONALE**

37382 None.

37383 **FUTURE DIRECTIONS**

37384 None.

37385 **SEE ALSO**

37386 The System Interfaces volume of IEEE Std. 1003.1-200x, *isatty()*, *ttyname()*

37387 **CHANGE HISTORY**

37388 First released in Issue 2.

37389 **Issue 4**

37390 Aligned with the ISO/IEC 9945-2: 1993 standard.

37391 **Issue 5**

37392 The SYNOPSIS is changed to indicate two forms of the command, with the second form marked  
37393 as obsolete. This is a clarification and does not change the functionality published in previous  
37394 issues.

37395 **NAME**37396           *type* — write a description of command *type*37397 **SYNOPSIS**37398 xsi       *type name...*

37399

37400 **DESCRIPTION**37401       The *type* utility shall indicate how each argument would be interpreted if used as a command  
37402       name.37403 **OPTIONS**

37404       None.

37405 **OPERANDS**

37406       The following operand shall be supported:

37407       *name*           A name to be interpreted.37408 **STDIN**

37409       Not used.

37410 **INPUT FILES**

37411       None.

37412 **ENVIRONMENT VARIABLES**37413       The following environment variables shall affect the execution of *type*:37414       *LANG*           Provide a default value for the internationalization variables that are unset or null.  
37415                       If *LANG* is unset or null, the corresponding value from the implementation-  
37416                       dependent default locale shall be used. If any of the internationalization variables  
37417                       contains an invalid setting, the utility shall behave as if none of the variables had  
37418                       been defined.37419       *LC\_ALL*          If set to a non-empty string value, override the values of all the other  
37420                       internationalization variables.37421       *LC\_CTYPE*       Determine the locale for the interpretation of sequences of bytes of text data as  
37422                       characters (for example, single-byte as opposed to multi-byte characters in  
37423                       arguments).37424       *LC\_MESSAGES*   Determine the locale that should be used to affect the format and contents of  
37425                       diagnostic messages written to standard error.  
3742637427       *NLSPATH*       Determine the location of message catalogs for the processing of *LC\_MESSAGES*.37428       *PATH*           Determine the location of *name*, as described in the System Interface Definitions  
37429                       volume of IEEE Std. 1003.1-200x, Chapter 8, Environment Variables.37430 **ASYNCHRONOUS EVENTS**

37431       Default.

37432 **STDOUT**37433       The standard output of *type* contains information about each operand in an unspecified format.  
37434       The information provided typically identifies the operand as a shell built-in, function, alias, or  
37435       keyword, and where applicable, may display the operand's path name.

37436 **STDERR**

37437           Used only for diagnostic messages.

37438 **OUTPUT FILES**

37439           None.

37440 **EXTENDED DESCRIPTION**

37441           None.

37442 **EXIT STATUS**

37443           The following exit values shall be returned:

37444           0   Successful completion.

37445           >0  An error occurred.

37446 **CONSEQUENCES OF ERRORS**

37447           Default.

37448 **APPLICATION USAGE**

37449           Since *type* must be aware of the contents of the current shell execution environment (such as the  
37450           lists of commands, functions, and built-ins processed by *hash*), it is always provided as a shell  
37451           regular built-in. If it is called in a separate utility execution environment, such as one of the  
37452           following:

37453           nohup type writer

37454           find . -type f | xargs type

37455           it might not produce accurate results.

37456 **EXAMPLES**

37457           None.

37458 **RATIONALE**

37459           None.

37460 **FUTURE DIRECTIONS**

37461           None.

37462 **SEE ALSO**

37463           *command*

37464 **CHANGE HISTORY**

37465           First released in Issue 2.

37466 **Issue 4**

37467           Relocated from the *sh* description to reflect its status as a regular built-in utility.



37468 **NAME**

37469           ulimit — set or report file size limit

37470 **SYNOPSIS**37471 xSI       ulimit [-f][*blocks*]

37472

37473 **DESCRIPTION**

37474       The *ulimit* utility shall set or report the file-size writing limit imposed on files written by the  
 37475 shell and its child processes (files of any size may be read). Only a process with appropriate  
 37476 privileges can increase the limit.

37477 **OPTIONS**

37478       The *ulimit* utility shall conform to the System Interface Definitions volume of  
 37479 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

37480       The following option shall be supported:

37481       -f           Set (or report, if no *blocks* operand is present), the file size limit in blocks. The -f  
 37482 option shall also be the default case.

37483 **OPERANDS**

37484       The following operand shall be supported:

37485       *blocks*       The number of 512-byte blocks to use as the new file size limit.

37486 **STDIN**

37487       Not used.

37488 **INPUT FILES**

37489       None.

37490 **ENVIRONMENT VARIABLES**

37491       The following environment variables shall affect the execution of *ulimit*:

37492       *LANG*       Provide a default value for the internationalization variables that are unset or null.  
 37493 If *LANG* is unset or null, the corresponding value from the implementation-  
 37494 dependent default locale shall be used. If any of the internationalization variables  
 37495 contains an invalid setting, the utility shall behave as if none of the variables had  
 37496 been defined.

37497       *LC\_ALL*      If set to a non-empty string value, override the values of all the other  
 37498 internationalization variables.

37499       *LC\_CTYPE*   Determine the locale for the interpretation of sequences of bytes of text data as  
 37500 characters (for example, single-byte as opposed to multi-byte characters in  
 37501 arguments).

37502       *LC\_MESSAGES*

37503                   Determine the locale that should be used to affect the format and contents of  
 37504 diagnostic messages written to standard error.

37505       *NLSPATH*    Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

37506 **ASYNCHRONOUS EVENTS**

37507       Default.

**37508 STDOUT**

37509       The standard output shall be used when no *blocks* operand is present. If the current number of  
37510       blocks is limited, the number of blocks in the current limit shall be written in the following  
37511       format:

37512       "%d\n", <number of 512-byte blocks>

37513       If there is no current limit on the number of blocks, in the POSIX locale the following format  
37514       shall be used:

37515       "unlimited\n"

**37516 STDERR**

37517       Used only for diagnostic messages.

**37518 OUTPUT FILES**

37519       None.

**37520 EXTENDED DESCRIPTION**

37521       None.

**37522 EXIT STATUS**

37523       The following exit values shall be returned:

37524       0   Successful completion.

37525       >0  A request for a higher limit was rejected or an error occurred.

**37526 CONSEQUENCES OF ERRORS**

37527       Default.

**37528 APPLICATION USAGE**

37529       Since *ulimit* affects the current shell execution environment, it is always provided as a shell  
37530       regular built-in. If it is called in separate utility execution environment, such as one of the  
37531       following:

37532       nohup ulimit -f 10000

37533       env ulimit 10000

37534       it does not affect the file size limit of the caller's environment.

37535       Once a limit has been decreased by a process, it cannot be increased (unless appropriate  
37536       privileges are involved), even back to the original system limit.

**37537 EXAMPLES**

37538       Set the file size limit to 51 200 bytes:

37539       ulimit -f 100

**37540 RATIONALE**

37541       None.

**37542 FUTURE DIRECTIONS**

37543       None.

**37544 SEE ALSO**

37545       The System Interfaces volume of IEEE Std. 1003.1-200x, *ulimit()*

**37546 CHANGE HISTORY**

37547       First released in Issue 2.

37548 **Issue 4**

37549 Relocated from the *sh* description to reflect its status as a regular built-in utility.

37550 **NAME**

37551           umask — get or set the file mode creation mask

37552 **SYNOPSIS**37553           umask [-S][*mask*]37554 **DESCRIPTION**

37555           The *umask* utility shall set the file mode creation mask of the current shell execution  
 37556           environment (see Section 2.12 on page 90) to the value specified by the *mask* operand. This mask  
 37557           shall affect the initial value of the file permission bits of subsequently created files. If *umask* is  
 37558           called in a subshell or separate utility execution environment, such as one of the following:

37559           (*umask* 002)37560           nohup *umask* ...37561           find . -exec *umask* ... \;

37562           it shall not affect the file mode creation mask of the caller's environment.

37563           If the *mask* operand is not specified, the *umask* utility shall write to standard output the value of  
 37564           the invoking process's file mode creation mask.

37565 **OPTIONS**

37566           The *umask* utility shall conform to the System Interface Definitions volume of  
 37567           IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

37568           The following option shall be supported:

37569           -S           Produce symbolic output.

37570           The default output style is unspecified, but shall be recognized on a subsequent invocation of  
 37571           *umask* on the same system as a *mask* operand to restore the previous file mode creation mask.

37572 **OPERANDS**

37573           The following operand shall be supported:

37574           *mask*           A string specifying the new file mode creation mask. The string is treated in the  
 37575           same way as the *mode* operand described in the the EXTENDED DESCRIPTION  
 37576           section for *chmod*.

37577           For a *symbolic\_mode* value, the new value of the file mode creation mask shall be  
 37578           the logical complement of the file permission bits portion of the file mode specified  
 37579           by the *symbolic\_mode* string.

37580           In a *symbolic\_mode* value, the permissions *op* characters '+' and '-' shall be  
 37581           interpreted relative to the current file mode creation mask; '+' shall cause the bits  
 37582           for the indicated permissions to be cleared in the mask; '-' shall cause the bits for  
 37583           the indicated permissions to be set in the mask.

37584           The interpretation of *mode* values that specify file mode bits other than the file  
 37585           permission bits is unspecified.

37586           MAN           In the octal integer form of *mode*, the specified bits are set in the file mode creation  
 37587           mask.

37588           The file mode creation mask shall be set to the resulting numeric value.

37589           The default output of a prior invocation of *umask* on the same system with no  
 37590           operand also shall be recognized as a *mask* operand. The use of an operand  
 37591           obtained in this way is not obsolescent, even if it is an octal number.

37592 **STDIN**

37593 Not used.

37594 **INPUT FILES**

37595 None.

37596 **ENVIRONMENT VARIABLES**37597 The following environment variables shall affect the execution of *umask*:

37598 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 37599 If *LANG* is unset or null, the corresponding value from the implementation-  
 37600 dependent default locale shall be used. If any of the internationalization variables  
 37601 contains an invalid setting, the utility shall behave as if none of the variables had  
 37602 been defined.

37603 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 37604 internationalization variables.

37605 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 37606 characters (for example, single-byte as opposed to multi-byte characters in  
 37607 arguments).

37608 *LC\_MESSAGES*

37609 Determine the locale that should be used to affect the format and contents of  
 37610 diagnostic messages written to standard error.

37611 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

37612 **ASYNCHRONOUS EVENTS**

37613 Default.

37614 **STDOUT**

37615 When the *mask* operand is not specified, the *umask* utility shall write a message to standard  
 37616 output that can later be used as a *umask mask* operand.

37617 If *-S* is specified, the message shall be in the following format:

37618 "u=%s,g=%s,o=%s\n", <owner permissions>, <group permissions>,  
 37619 <other permissions>

37620 where the three values shall be combinations of letters from the set {r, w, x}; the presence of a  
 37621 letter shall indicate that the corresponding bit is clear in the file mode creation mask.

37622 If a *mask* operand is specified, there shall be no output written to standard output.37623 **STDERR**

37624 Used only for diagnostic messages.

37625 **OUTPUT FILES**

37626 None.

37627 **EXTENDED DESCRIPTION**

37628 None.

37629 **EXIT STATUS**

37630 The following exit values shall be returned:

37631 0 The file mode creation mask was successfully changed, or no *mask* operand was supplied.

37632 &gt;0 An error occurred.

## 37633 CONSEQUENCES OF ERRORS

37634 Default.

## 37635 APPLICATION USAGE

37636 Since *umask* affects the current shell execution environment, it is generally provided as a shell  
 37637 regular built-in.

37638 In contrast to the negative permission logic provided by the file mode creation mask and the  
 37639 octal number form of the *mask* argument, the symbolic form of the *mask* argument specifies those  
 37640 permissions that are left alone.

## 37641 EXAMPLES

37642 Either of the commands:

37643 `umask a=rx,ug+w`37644 `umask 002`

37645 sets the mode mask so that subsequently created files have their S\_IWOTH bit cleared.

37646 After setting the mode mask with either of the above commands, the *umask* command can be  
 37647 used to write out the current value of the mode mask:

37648 `$ umask`37649 `0002`

37650 (The output format is unspecified, but historical implementations use the obsolescent octal  
 37651 integer mode format.)

37652 `$ umask -S`37653 `u=rwx,g=rwx,o=rx`

37654 Either of these outputs can be used as the mask operand to a subsequent invocation of the *umask*  
 37655 utility.

37656 Assuming the mode mask is set as above, the command:

37657 `umask g-w`

37658 sets the mode mask so that subsequently created files have their S\_IWGRP and S\_IWOTH bits  
 37659 cleared.

37660 The command:

37661 `umask -- -w`

37662 sets the mode mask so that subsequently created files have all their write bits cleared. Note that  
 37663 *mask* operands `-r`, `-w`, `-x` or anything beginning with a hyphen, must be preceded by `--` to  
 37664 keep it from being interpreted as an option.

## 37665 RATIONALE

37666 Since *umask* affects the current shell execution environment, it is generally provided as a shell  
 37667 regular built-in. If it is called in a subshell or separate utility execution environment, such as one  
 37668 of the following:

37669 `(umask 002)`37670 `nohup umask ...`37671 `find . -exec umask ... \;`

37672 it does not affect the file mode creation mask of the environment of the caller.

37673 The description of the historical utility was modified to allow it to use the symbolic modes of  
 37674 *chmod*. The `-s` option used in early proposals was changed to `-S` because `-s` could be confused

37675 with a *symbolic\_mode* form of mask referring to the S\_ISUID and S\_ISGID bits.

37676 **Notes to Reviewers**

37677 *This section with side shading will not appear in the final copy. - Ed.*

37678 D1, XCU, ERN 355 suggests we should specify the default output. Suggestions please.

37679 The default output style is implementation-dependent to permit implementors to provide  
37680 migration to the new symbolic style at the time most appropriate to their users. An **-o** flag to  
37681 force octal mode output was omitted because the octal mode may not be sufficient to specify all  
37682 of the information that may be present in the file mode creation mask when more secure file  
37683 access permission checks are implemented.

37684 It has been suggested that trusted systems developers might appreciate ameliorating the  
37685 requirement that the mode mask “affects” the file access permissions, since it seems access  
37686 control lists might replace the mode mask to some degree. The wording has been changed to say  
37687 that it affects the file permission bits, and it leaves the details of the behavior of how they affect  
37688 the file access permissions to the description in the System Interfaces volume of  
37689 IEEE Std. 1003.1-200x.

37690 **FUTURE DIRECTIONS**

37691 None.

37692 **SEE ALSO**

37693 *chmod*, the System Interfaces volume of IEEE Std. 1003.1-200x, *umask()*

37694 **CHANGE HISTORY**

37695 First released in Issue 2.

37696 **Issue 4**

37697 Aligned with the ISO/IEC 9945-2:1993 standard.

37698 **Issue 6**

37699 The following new requirements on POSIX implementations derive from alignment with the  
37700 Single UNIX Specification:

- 37701
- The octal mode is supported.

37702 **NAME**

37703 unalias — remove alias definitions

37704 **SYNOPSIS**37705 UP unalias *alias-name*...

37706 unalias -a

37707

37708 **DESCRIPTION**

37709 The *unalias* utility shall remove the definition for each alias name specified. See Section 2.3.1 on  
 37710 page 40. The aliases shall be removed from the current shell execution environment; see Section  
 37711 2.12 on page 90.

37712 **OPTIONS**

37713 The *unalias* utility shall conform to the System Interface Definitions volume of  
 37714 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

37715 The following option shall be supported:

37716 **-a** Remove all alias definitions from the current shell execution environment.

37717 **OPERANDS**

37718 The following operand shall be supported:

37719 *alias-name* The name of an alias to be removed.

37720 **STDIN**

37721 Not used.

37722 **INPUT FILES**

37723 None.

37724 **ENVIRONMENT VARIABLES**

37725 The following environment variables shall affect the execution of *unalias*:

37726 **LANG** Provide a default value for the internationalization variables that are unset or null.  
 37727 If *LANG* is unset or null, the corresponding value from the implementation-  
 37728 dependent default locale shall be used. If any of the internationalization variables  
 37729 contains an invalid setting, the utility shall behave as if none of the variables had  
 37730 been defined.

37731 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 37732 internationalization variables.

37733 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 37734 characters (for example, single-byte as opposed to multi-byte characters in  
 37735 arguments).

37736 **LC\_MESSAGES**

37737 Determine the locale that should be used to affect the format and contents of  
 37738 diagnostic messages written to standard error.

37739 **XS1** **NLS\_PATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

37740 **ASYNCHRONOUS EVENTS**

37741 Default.



37742 **STDOUT**

37743 Not used.

37744 **STDERR**

37745 Used only for diagnostic messages.

37746 **OUTPUT FILES**

37747 None.

37748 **EXTENDED DESCRIPTION**

37749 None.

37750 **EXIT STATUS**

37751 The following exit values shall be returned:

37752 0 Successful completion.

37753 >0 One of the *alias-name* operands specified did not represent a valid alias definition, or an  
37754 error occurred.37755 **CONSEQUENCES OF ERRORS**

37756 Default.

37757 **APPLICATION USAGE**37758 Since *unalias* affects the current shell execution environment, it is generally provided as a shell  
37759 regular built-in.37760 Application writers should note that this utility need not be provided on systems that do not  
37761 support the User Portability Utilities option.37762 **EXAMPLES**

37763 None.

37764 **RATIONALE**37765 The *unalias* description is based on that from historical KornShell implementations. Known  
37766 differences exist between that and the C shell. The KornShell version was adopted to be  
37767 consistent with all the other KornShell features in this volume of IEEE Std. 1003.1-200x, such as  
37768 command line editing.37769 The *-a* option is the equivalent of the *unalias\** form of the C shell and is provided to address  
37770 security concerns about unknown aliases entering the environment of a user (or application)  
37771 through the allowable implementation-dependent predefined alias route or as a result of an *ENV*  
37772 file. (Although *unalias* could be used to simplify the “secure” shell script shown in the *command*  
37773 rationale, it does not obviate the need to quote all command names. An initial call to *unalias -a*  
37774 would have to be quoted in case there was an alias for *unalias*.)37775 **FUTURE DIRECTIONS**

37776 None.

37777 **SEE ALSO**37778 *alias*37779 **CHANGE HISTORY**

37780 First released in Issue 4.

37781 **Issue 6**

37782 This utility is now marked as part of the User Portability Utilities option.

37783 **NAME**37784            *uname* — return system name37785 **SYNOPSIS**37786            *uname* [*-snrvma*]37787 **DESCRIPTION**

37788            By default, the *uname* utility shall write the operating system name to standard output. When  
 37789            options are specified, symbols representing one or more system characteristics shall be written  
 37790            to the standard output. The format and contents of the symbols are implementation-dependent.  
 37791            On systems conforming to the System Interfaces volume of IEEE Std. 1003.1-200x, the symbols  
 37792            written shall be those supported by the *uname()* function as defined in the System Interfaces  
 37793            volume of IEEE Std. 1003.1-200x.

37794 **OPTIONS**

37795            The *uname* utility shall conform to the System Interface Definitions volume of  
 37796            IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

37797            The following options shall be supported:

- 37798            **-a**            Behave as though all of the options **-mnrsv** were specified.
- 37799            **-m**            Write the name of the hardware type on which the system is running to standard  
 37800            output.
- 37801            **-n**            Write the name of this node within an implementation-dependent  
 37802            communications network.
- 37803            **-r**            Write the current release level of the operating system implementation.
- 37804            **-s**            Write the name of the implementation of the operating system.
- 37805            **-v**            Write the current version level of this release of the operating system  
 37806            implementation.
- 37807            If no options are specified, the *uname* utility shall write the operating system name, as if the **-s**  
 37808            option had been specified.

37809 **OPERANDS**

37810            None.

37811 **STDIN**

37812            Not used.

37813 **INPUT FILES**

37814            None.

37815 **ENVIRONMENT VARIABLES**37816            The following environment variables shall affect the execution of *uname*:

- 37817            **LANG**        Provide a default value for the internationalization variables that are unset or null.  
 37818            If **LANG** is unset or null, the corresponding value from the implementation-  
 37819            dependent default locale shall be used. If any of the internationalization variables  
 37820            contains an invalid setting, the utility shall behave as if none of the variables had  
 37821            been defined.
- 37822            **LC\_ALL**        If set to a non-empty string value, override the values of all the other  
 37823            internationalization variables.
- 37824            **LC\_CTYPE**     Determine the locale for the interpretation of sequences of bytes of text data as  
 37825            characters (for example, single-byte as opposed to multi-byte characters in

arguments).

**LC\_MESSAGES**  
Determine the locale that should be used to affect the format and contents of diagnostic messages written to standard error.

**NLSPATH** Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

**ASYNCHRONOUS EVENTS**  
Default.

**STDOUT**  
By default, the output shall be a single line of the following form:  
"%s\n", <sysname>  
If the **-a** option is specified, the output shall be a single line of the following form:  
"%s %s %s %s %s\n", <sysname>, <nodename>, <release>, <version>, <machine>  
Additional implementation-dependent symbols may be written; all such symbols shall be written at the end of the line of output before the <newline> character.  
If options are specified to select different combinations of the symbols, only those symbols shall be written, in the order shown above for the **-a** option. If a symbol is not selected for writing, its corresponding trailing <blank> characters also shall not be written.

**STDERR**  
Used only for diagnostic messages.

**OUTPUT FILES**  
None.

**EXTENDED DESCRIPTION**  
None.

**EXIT STATUS**  
The following exit values shall be returned:  
0 The requested information was successfully written.  
>0 An error occurred.

**CONSEQUENCES OF ERRORS**  
Default.

**APPLICATION USAGE**  
Note that any of the symbols could include embedded <space> characters, which may affect parsing algorithms if multiple options are selected for output.  
The node name is typically a name that the system uses to identify itself for intersystem communication addressing.

**EXAMPLES**  
The following command:  
uname -sr  
writes the operating system name and release level, separated by one or more <blank> characters.

**37866 RATIONALE**

37867 It was suggested that this utility cannot be used portably since the format of the symbols is  
37868 implementation-dependent. The POSIX.1 working group could not achieve consensus on  
37869 defining these formats in the underlying *uname()* function, and there was no expectation that  
37870 this volume of IEEE Std. 1003.1-200x would be any more successful. Some applications may still  
37871 find this historical utility of value. For example, the symbols could be used for system log entries  
37872 or for comparison with operator or user input.

**37873 FUTURE DIRECTIONS**

37874 None.

**37875 SEE ALSO**

37876 The System Interfaces volume of IEEE Std. 1003.1-200x, *uname()*

**37877 CHANGE HISTORY**

37878 First released in Issue 2.

**37879 Issue 4**

37880 Aligned with the ISO/IEC 9945-2:1993 standard.

**37881 Issue 4, Version 2**

37882 The SYNOPSIS section lists all the valid options.

37883 **NAME**

37884 uncompress — expand compressed data

37885 **SYNOPSIS**37886 xSI uncompress [-cfv][*file...*]

37887

37888 **DESCRIPTION**

37889 The *uncompress* utility shall restore files to their original state after they have been compressed  
 37890 using the *compress* utility. If no files are specified, the standard input shall be uncompressed to  
 37891 the standard output. If the invoking process has appropriate privileges, the ownership, modes,  
 37892 access time, and modification time of the original file shall be preserved.

37893 This utility shall support the uncompressing of any files produced by the *compress* utility on the  
 37894 same implementation. For files produced by *compress* on other systems, *uncompress* supports 9 to  
 37895 14-bit compression (see *compress* on page 299, -b); it is implementation-dependent whether  
 37896 values of -b greater than 14 are supported.

37897 **OPTIONS**

37898 The *uncompress* utility shall conform to the System Interface Definitions volume of  
 37899 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

37900 The following options shall be supported:

- 37901 -c Write to standard output; no files are changed.
- 37902 -f Do not prompt for overwriting files. Except when run in the background, if -f is  
 37903 not given the user shall be prompted as to whether an existing file should be  
 37904 overwritten. If the standard input is not a terminal and -f is not given, *uncompress*  
 37905 shall write a diagnostic message to standard error and exit with a status greater  
 37906 than zero.
- 37907 -v Write messages to standard error concerning the expansion of each file.

37908 **OPERANDS**

37909 The following operand shall be supported:

- 37910 *file* A path name of a file. If *file* already has the .Z suffix specified, it shall be used as  
 37911 the input file and the output file shall be named *file* with the .Z suffix removed.  
 37912 Otherwise, *file* shall be used as the name of the output file and *file* with the .Z  
 37913 suffix appended shall be used as the input file.

37914 **STDIN**

37915 The standard input shall be used only if no *file* operands are specified, or if a *file* operand is '- '.

37916 **INPUT FILES**

37917 Input files shall be in the format produced by the *compress* utility.

37918 **ENVIRONMENT VARIABLES**

37919 The following environment variables shall affect the execution of *uncompress*:

- 37920 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 37921 If *LANG* is unset or null, the corresponding value from the implementation-  
 37922 dependent default locale shall be used. If any of the internationalization variables  
 37923 contains an invalid setting, the utility shall behave as if none of the variables had  
 37924 been defined.
- 37925 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 37926 internationalization variables.

- 37927        *LC\_CTYPE*   Determine the locale for the interpretation of sequences of bytes of text data as  
37928                   characters (for example, single-byte as opposed to multi-byte characters in  
37929                   arguments).
- 37930        *LC\_MESSAGES*  
37931                   Determine the locale that should be used to affect the format and contents of  
37932                   diagnostic messages written to standard error.
- 37933        *NLSPATH*   Determine the location of message catalogs for the processing of *LC\_MESSAGES*.
- 37934 **ASYNCHRONOUS EVENTS**  
37935        Default.
- 37936 **STDOUT**  
37937        When there are no file operands or the *-c* option is specified, the uncompressed output is  
37938        written to standard output.
- 37939 **STDERR**  
37940        Prompts shall be written to the standard error output under the conditions specified in the  
37941        DESCRIPTION and OPTIONS sections. The prompts shall contain the *file* path name, but their  
37942        format is otherwise unspecified. Otherwise, the standard error output shall be used only for  
37943        diagnostic messages.
- 37944 **OUTPUT FILES**  
37945        Output files are the same as the respective input files to *compress*.
- 37946 **EXTENDED DESCRIPTION**  
37947        None.
- 37948 **EXIT STATUS**  
37949        The following exit values shall be returned:  
37950        0   Successful completion.  
37951        >0   An error occurred.
- 37952 **CONSEQUENCES OF ERRORS**  
37953        The input file remains unmodified.
- 37954 **APPLICATION USAGE**  
37955        The limit of 14 on the *compress -b bits* argument is to achieve portability to all systems (within  
37956        the restrictions imposed by the lack of an explicit published file format). Some systems based on  
37957        16-bit architectures cannot support 15 or 16-bit uncompression.
- 37958 **EXAMPLES**  
37959        None.
- 37960 **RATIONALE**  
37961        None.
- 37962 **FUTURE DIRECTIONS**  
37963        None.
- 37964 **SEE ALSO**  
37965        *compress*, *zcat*
- 37966 **CHANGE HISTORY**  
37967        First released in Issue 4.

37968 **Issue 4, Version 2**

37969       The DESCRIPTION is clarified to state that the ownership, modes, access time, and modification  
37970       time of the original file are preserved if the invoking process has appropriate privileges.

37971 **Issue 6**

37972       The normative text is reworded to avoid use of the term “must” for application requirements.

37973 **NAME**

37974 unexpand — convert spaces to tabs

37975 **SYNOPSIS**37976 UP unexpand [ -a | -t *tablist* ][*file...*]

37977

37978 **DESCRIPTION**

37979 The *unexpand* utility shall copy files or standard input to standard output, converting <blank>  
 37980 characters at the beginning of each line into the maximum number of <tab> characters followed  
 37981 by the minimum number of <space> characters needed to fill the same column positions  
 37982 originally filled by the translated <blank> characters. By default, tabstops shall be set at every  
 37983 eighth column position. Each <backspace> character shall be copied to the output, and shall  
 37984 cause the column position count for tab calculations to be decremented; the count shall never be  
 37985 decremented to a value less than one.

37986 **OPTIONS**

37987 The *unexpand* utility shall conform to the System Interface Definitions volume of  
 37988 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

37989 The following options shall be supported:

37990 **-a** In addition to translating <blank> characters at the beginning of each line, translate  
 37991 all sequences of two or more <blank> characters immediately preceding a tab stop  
 37992 to the maximum number of <tab> characters followed by the minimum number of  
 37993 <space> characters needed to fill the same column positions originally filled by the  
 37994 translated <blank> characters.

37995 **-t *tablist*** Specify the tab stops. The application shall ensure that the *tablist* option-argument  
 37996 is a single argument consisting of a single positive decimal integer or multiple  
 37997 positive decimal integers, separated by <blank> characters or commas, in  
 37998 ascending order. If a single number is given, tabs shall be set *tablist* column  
 37999 positions apart instead of the default 8. If multiple numbers are given, the tabs  
 38000 shall be set at those specific column positions.

38001 The application shall ensure that each tab-stop position *N* is an integer value  
 38002 greater than zero, and the list shall be in strictly ascending order. This is taken to  
 38003 mean that, from the start of a line of output, tabbing to position *N* shall cause the  
 38004 next character output to be in the (*N*+1)th column position on that line. When the  
 38005 **-t** option is not specified, the default shall be the equivalent of specifying **-t 8**  
 38006 (except for the interaction with **-a**, described below).

38007 No <space>-to-<tab> character conversions shall occur for characters at positions  
 38008 beyond the last of those specified in a multiple tab-stop list.

38009 When **-t** is specified, the presence or absence of the **-a** option shall be ignored;  
 38010 conversion shall not be limited to the processing of leading <blank> characters.

38011 **OPERANDS**

38012 The following operand shall be supported:

38013 ***file*** A path name of a text file to be used as input.

38014 **STDIN**

38015 See the INPUT FILES section.



38016 **INPUT FILES**

38017       The input files shall be text files.

38018 **ENVIRONMENT VARIABLES**38019       The following environment variables shall affect the execution of *unexpand*:

38020       **LANG**       Provide a default value for the internationalization variables that are unset or null.  
 38021                   If *LANG* is unset or null, the corresponding value from the implementation-  
 38022                   dependent default locale shall be used. If any of the internationalization variables  
 38023                   contains an invalid setting, the utility shall behave as if none of the variables had  
 38024                   been defined.

38025       **LC\_ALL**       If set to a non-empty string value, override the values of all the other  
 38026                   internationalization variables.

38027       **LC\_CTYPE**   Determine the locale for the interpretation of sequences of bytes of text data as  
 38028                   characters (for example, single-byte as opposed to multi-byte characters in  
 38029                   arguments and input files), the processing of <tab> and <space> characters and for  
 38030                   the determination of the width in column positions each character would occupy  
 38031                   on an output device.

38032       **LC\_MESSAGES**

38033                   Determine the locale that should be used to affect the format and contents of  
 38034                   diagnostic messages written to standard error.

38035 **XSI**       **NLSPATH**   Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

38036 **ASYNCHRONOUS EVENTS**

38037       Default.

38038 **STDOUT**

38039       The standard output is equivalent to the input files with the specified <space>-to-<tab>  
 38040       character conversions.

38041 **STDERR**

38042       Used only for diagnostic messages.

38043 **OUTPUT FILES**

38044       None.

38045 **EXTENDED DESCRIPTION**

38046       None.

38047 **EXIT STATUS**

38048       The following exit values shall be returned:

38049       0   Successful completion.

38050       &gt;0   An error occurred.

38051 **CONSEQUENCES OF ERRORS**

38052       Default.

**38053 APPLICATION USAGE**

38054 One non-intuitive aspect of *unexpand* is its restriction to leading spaces when neither **-a** nor **-t** is  
38055 specified. Users who desire to always convert all spaces in a file can easily alias *unexpand* to use  
38056 the **-a** or **-t 8** option.

38057 Application writers should note that this utility need not be provided on systems that do not  
38058 support the User Portability Utilities option.

**38059 EXAMPLES**

38060 None.

**38061 RATIONALE**

38062 On several occasions, consideration was given to adding a **-t** option to the *unexpand* utility to  
38063 complement the **-t** in *expand* (see *expand* on page 460). The historical intent of *unexpand* was to  
38064 translate multiple <blank>s into tab stops, where tab stops were a multiple of eight column  
38065 positions on most UNIX systems. An early proposal omitted **-t** because it seemed outside the  
38066 scope of the UPE; it was not described in any of the base documents. However, hard-coding tab  
38067 stops every eight columns was not suitable for the international community and broke historical  
38068 precedents for some vendors in the FORTRAN community, so **-t** was restored in conjunction  
38069 with the list of valid extension categories considered by the standard developers. Thus, *unexpand*  
38070 is now the logical converse of *expand*.

**38071 FUTURE DIRECTIONS**

38072 None.

**38073 SEE ALSO**

38074 *expand*, *tabs*

**38075 CHANGE HISTORY**

38076 First released in Issue 4.

**38077 Issue 6**

38078 This utility is now marked as part of the User Portability Utilities option. |

38079 The definition of the *LC\_CTYPE* environment variable is changed to align with the |  
38080 IEEE P1003.2b draft standard. |

38081 The normative text is reworded to avoid use of the term “must” for application requirements. |

38082 **NAME**38083           unget — undo a previous get of an SCCS file (**DEVELOPMENT**)38084 **SYNOPSIS**38085 xSI       unget [-ns][-r *SID*] *file...*

38086

38087 **DESCRIPTION**38088       The *unget* utility shall reverse the effect of a *get -e* done prior to creating the intended new delta.38089 **OPTIONS**38090       The *unget* utility shall conform to the System Interface Definitions volume of  
38091       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

38092       The following options shall be supported:

38093       -r *SID*       Uniquely identify which delta is no longer intended. (This would have been  
38094                      specified by *get* as the new delta.) The use of this option is necessary only if two or  
38095                      more outstanding *get* commands for editing on the same SCCS file were done by  
38096                      the same person (login name).

38097       -s           Suppress the writing to standard output of the intended delta's SID.

38098       -n           Retain the file that was obtained by *get*, which would normally be removed from  
38099                      the current directory.38100 **OPERANDS**

38101       The following operands shall be supported:

38102       *file*        A path name of an existing SCCS file or a directory. If *file* is a directory, *unget*  
38103                      behaves as though each file in the directory were specified as a named file, except  
38104                      that non-SCCS files (last component of the path name does not begin with *s.*) and  
38105                      unreadable files shall be silently ignored.38106                      If a single instance *file* is specified as '-', the standard input shall be read; each  
38107                      line of the standard input is taken to be the name of an SCCS file to be processed.  
38108                      Non-SCCS files and unreadable files shall be silently ignored.38109 **STDIN**38110       The standard input shall be a text file used only when the *file* operand is specified as '-'. Each  
38111       line of the text file shall be interpreted as an SCCS path name.38112 **INPUT FILES**

38113       Any SCCS files processed are files of an unspecified format.

38114 **ENVIRONMENT VARIABLES**38115       The following environment variables shall affect the execution of *unget*:38116       LANG        Provide a default value for the internationalization variables that are unset or null.  
38117                      If *LANG* is unset or null, the corresponding value from the implementation-  
38118                      dependent default locale shall be used. If any of the internationalization variables  
38119                      contains an invalid setting, the utility shall behave as if none of the variables had  
38120                      been defined.38121       LC\_ALL       If set to a non-empty string value, override the values of all the other  
38122                      internationalization variables.38123       LC\_CTYPE     Determine the locale for the interpretation of sequences of bytes of text data as  
38124                      characters (for example, single-byte as opposed to multi-byte characters in  
38125                      arguments and input files).

38126 **LC\_MESSAGES**

38127 Determine the locale that should be used to affect the format and contents of  
38128 diagnostic messages written to standard error.

38129 **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

38130 **ASYNCHRONOUS EVENTS**

38131 Default.

38132 **STDOUT**

38133 The standard output shall consist of a line for each file, in the following format:

38134 "%s\n", <*SID removed from file*>

38135 If there is more than one named file or if a directory or standard input is named, each path name  
38136 shall be written before each of the preceding lines:

38137 "\n%s:\n", <*pathname*>

38138 **STDERR**

38139 Used only for diagnostic messages.

38140 **OUTPUT FILES**

38141 Any SCCS files updated are files of an unspecified format. During processing of a *file*, a locking  
38142 *z-file*, as described in *get*, and a *q-file* (a working copy of the *p-file*), may be created and deleted.  
38143 The *p-file* and *g-file*, as described in *get*, shall be deleted.

38144 **EXTENDED DESCRIPTION**

38145 None.

38146 **EXIT STATUS**

38147 The following exit values shall be returned:

38148 0 Successful completion.

38149 >0 An error occurred.

38150 **CONSEQUENCES OF ERRORS**

38151 Default.

38152 **APPLICATION USAGE**

38153 None.

38154 **EXAMPLES**

38155 None.

38156 **RATIONALE**

38157 None.

38158 **FUTURE DIRECTIONS**

38159 None.

38160 **SEE ALSO**

38161 *delta*, *get*, *sact*

38162 **CHANGE HISTORY**

38163 First released in Issue 2.

38164 **Issue 4**

38165 Format reorganized.

38166 Utility Syntax Guidelines support mandated.

|       |                                                                                              |  |
|-------|----------------------------------------------------------------------------------------------|--|
| 38167 | Internationalized environment variable support mandated.                                     |  |
| 38168 | <b>Issue 6</b>                                                                               |  |
| 38169 | The normative text is reworded to avoid use of the term “must” for application requirements. |  |

38170 **NAME**

38171           uniq — report or filter out repeated lines in a file

38172 **SYNOPSIS**38173           uniq [-c|-d|-u][-f *fields*][-s *char*][*input\_file* [*output\_file*]]38174 **DESCRIPTION**

38175           The *uniq* utility shall read an input file comparing adjacent lines, and writes one copy of each  
 38176           input line on the output. The second and succeeding copies of repeated adjacent input lines shall  
 38177           not be written.

38178           Repeated lines in the input shall not be detected if they are not adjacent.

38179 **OPTIONS**

38180           The *uniq* utility shall conform to the System Interface Definitions volume of  
 38181           IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

38182           The following options shall be supported:

38183           -c           Precede each output line with a count of the number of times the line occurred in  
 38184           the input.

38185           -d           Suppress the writing of lines that are not repeated in the input.

38186           -f *fields*   Ignore the first *fields* fields on each input line when doing comparisons, where  
 38187           *fields* is a positive decimal integer. A field is the maximal string matched by the  
 38188           basic regular expression:

38189           [[[:blank:]]]\*^[[:blank:]]\*

38190           If the *fields* option-argument specifies more fields than appear on an input line, a  
 38191           null string shall be used for comparison.

38192           -s *chars*   Ignore the first *chars* characters when doing comparisons, where *chars* shall be a  
 38193           positive decimal integer. If specified in conjunction with the -f option, the first  
 38194           *chars* characters after the first *fields* fields shall be ignored. If the *chars* option-  
 38195           argument specifies more characters than remain on an input line, a null string shall  
 38196           be used for comparison.

38197           -u           Suppress the writing of lines that are repeated in the input.

38198 **OPERANDS**

38199           The following operands shall be supported:

38200           *input\_file*   A path name of the input file. If the *input\_file* operand is not specified, or if the  
 38201           *input\_file* is '-', the standard input is used.

38202           *output\_file*   A path name of the output file. If the *output\_file* operand is not specified, the  
 38203           standard output shall be used. The results are unspecified if the file named by  
 38204           *output\_file* is the file named by *input\_file*.

38205 **STDIN**

38206           The standard input shall be used only if no *input\_file* operand is specified or if *input\_file* is '-'.  
 38207           See the INPUT FILES section.

38208 **INPUT FILES**

38209           The input file shall be a text file.

**38210 ENVIRONMENT VARIABLES**

38211 The following environment variables shall affect the execution of *uniq*:

38212 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 38213 If *LANG* is unset or null, the corresponding value from the implementation-  
 38214 dependent default locale shall be used. If any of the internationalization variables  
 38215 contains an invalid setting, the utility shall behave as if none of the variables had  
 38216 been defined.

38217 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 38218 internationalization variables.

38219 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 38220 characters (for example, single-byte as opposed to multi-byte characters in  
 38221 arguments and input files) which characters constitute a <blank> character in the  
 38222 current locale.

38223 *LC\_MESSAGES*

38224 Determine the locale that should be used to affect the format and contents of  
 38225 diagnostic messages written to standard error.

38226 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

**38227 ASYNCHRONOUS EVENTS**

38228 Default.

**38229 STDOUT**

38230 The standard output shall be used only if no *output\_file* operand is specified. See the OUTPUT  
 38231 FILES section.

**38232 STDERR**

38233 Used only for diagnostic messages.

**38234 OUTPUT FILES**

38235 If the *-c* option is specified, the application shall ensure that the output file is empty or each line  
 38236 shall be of the form:

38237 "%d %s", <number of duplicates>, <line>

38238 otherwise, the application shall ensure that the output file is empty or each line shall be of the  
 38239 form:

38240 "%s", <line>

**38241 EXTENDED DESCRIPTION**

38242 None.

**38243 EXIT STATUS**

38244 The following exit values shall be returned:

38245 0 The utility executed successfully.

38246 >0 An error occurred.

**38247 CONSEQUENCES OF ERRORS**

38248 Default.

## 38249 APPLICATION USAGE

38250 The *sort* utility can be used to cause repeated lines to be adjacent in the input file.

## 38251 EXAMPLES

38252 The following input file data (but flushed left) was used for a test series on *uniq*:

```
38253 #01 foo0 bar0 fool bar1
38254 #02 bar0 fool bar1 fool
38255 #03 foo0 bar0 fool bar1
38256 #04
38257 #05 foo0 bar0 fool bar1
38258 #06 foo0 bar0 fool bar1
38259 #07 bar0 fool bar1 foo0
```

38260 What follows is a series of test invocations of the *uniq* utility that use a mixture of *uniq* options  
38261 against the input file data. These tests verify the meaning of *adjacent*. The *uniq* utility views the  
38262 input data as a sequence of strings delimited by '\n'. Accordingly, for the *fieldsth* member of  
38263 the sequence, *uniq* interprets unique or repeated adjacent lines strictly relative to the *fields+1*th  
38264 member.

- 38265 1. This first example tests the line counting option, comparing each line of the input file data  
38266 starting from the second field:

```
38267 uniq -c -f 1 uniq_0I.t
38268 1 #01 foo0 bar0 fool bar1
38269 1 #02 bar0 fool bar1 foo0
38270 1 #03 foo0 bar0 fool bar1
38271 1 #04
38272 2 #05 foo0 bar0 fool bar1
38273 1 #07 bar0 fool bar1 foo0
```

38274 The number '2', prefixing the fifth line of output, signifies that the *uniq* utility detected a  
38275 pair of repeated lines. Given the input data, this can only be true when *uniq* is run using  
38276 the *-f 1* option (which shall cause *uniq* to ignore the first field on each input line).

- 38277 2. The second example tests the option to suppress unique lines, comparing each line of the  
38278 input file data starting from the second field:

```
38279 uniq -d -f 1 uniq_0I.t
38280 #05 foo0 bar0 fool bar1
```

- 38281 3. This test suppresses repeated lines, comparing each line of the input file data starting from  
38282 the second field:

```
38283 uniq -u -f 1 uniq_0I.t
38284 #01 foo0 bar0 fool bar1
38285 #02 bar0 fool bar1 fool
38286 #03 foo0 bar0 fool bar1
38287 #04
38288 #07 bar0 fool bar1 foo0
```

- 38289 4. This suppresses unique lines, comparing each line of the input file data starting from the  
38290 third character:

```
38291 uniq -d -s 2 uniq_0I.t
```

38292 In the last example, the *uniq* utility found no input matching the above criteria.



38293 **RATIONALE**

38294           Some historical implementations have limited lines to be 1 080 bytes in length, which does not  
38295           meet the implied {LINE\_MAX} limit.

38296           The **-f** and **-s** options were added to replace the obsolescent **-n** and **+m** options so that *uniq*  
38297           could meet the syntax guidelines in an upward-compatible way.

38298 **FUTURE DIRECTIONS**

38299           None.

38300 **SEE ALSO**

38301           *comm*, *sort*

38302 **CHANGE HISTORY**

38303           First released in Issue 2.

38304 **Issue 4**

38305           Aligned with the ISO/IEC 9945-2: 1993 standard.

38306 **Issue 6**

38307           The obsolescent SYNOPSIS and associated text are removed. |

38308           The normative text is reworded to avoid use of the term “must” for application requirements. |

38309 **NAME**

38310 unlink — call the *unlink()* function

38311 **SYNOPSIS**

38312 xSI unlink *file*

38313

38314 **DESCRIPTION**

38315 The *unlink* utility shall perform the function call:

38316 unlink(*file*);

38317 A user may need appropriate privilege to invoke the *unlink* utility.

38318 **OPTIONS**

38319 None.

38320 **OPERANDS**

38321 The following operands shall be supported:

38322 *file* The path name of an existing file.

38323 **STDIN**

38324 Not used.

38325 **INPUT FILES**

38326 Not used.

38327 **ENVIRONMENT VARIABLES**

38328 The following environment variables shall affect the execution of *unlink*:

38329 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 38330 If *LANG* is unset or null, the corresponding value from the implementation-  
 38331 dependent default locale shall be used. If any of the internationalization variables  
 38332 contain an invalid setting, the utility behaves as if none of the variables had been  
 38333 set.

38334 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 38335 internationalization variables.

38336 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 38337 characters (for example, single-byte as opposed to multi-byte characters in  
 38338 arguments).

38339 *LC\_MESSAGES*  
 38340 Determine the locale that should be used to affect the format and contents of  
 38341 diagnostic messages written to standard error.

38342 *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

38343 **ASYNCHRONOUS EVENTS**

38344 Default.

38345 **STDOUT**

38346 None.

38347 **STDERR**

38348 Used only for diagnostic messages.

**38349 OUTPUT FILES**

38350 None.

**38351 EXTENDED DESCRIPTION**

38352 None.

**38353 EXIT STATUS**

38354 The following exit values shall be returned:

38355 0 Successful completion.

38356 >0 An error occurred.

**38357 CONSEQUENCES OF ERRORS**

38358 Default.

**38359 APPLICATION USAGE**

38360 None.

**38361 EXAMPLES**

38362 None.

**38363 RATIONALE**

38364 None.

**38365 FUTURE DIRECTIONS**

38366 None.

**38367 SEE ALSO**

38368 *link*, *rm*, the System Interfaces volume of IEEE Std. 1003.1-200x, *unlink()*

**38369 CHANGE HISTORY**

38370 First released in Issue 5.

## 38371 NAME

38372 uucp — system-to-system copy

## 38373 SYNOPSIS

38374 UN XSI uucp [-cCdfjmr][*-n user*] *source-file... destination-file*

38375

## 38376 DESCRIPTION

38377 The *uucp* utility shall copy files named by the *source-file* arguments to the *destination-file*  
38378 argument. The files named can be on local or remote systems.38379 The *uucp* utility cannot guarantee support for all character encodings in all circumstances. For  
38380 example, transmission data may be restricted to 7 bits by the underlying network, 8-bit data and  
38381 file names need not be portable to non-internationalized systems, and so on. Under these  
38382 circumstances, it is recommended that only characters defined in the ISO/IEC 646:1991  
38383 standard International Reference Version (equivalent to ASCII) 7-bit range of characters be used,  
38384 and that only characters defined in the Portable File Name Character Set be used for naming  
38385 files.

## 38386 OPTIONS

38387 The *uucp* utility shall conform to the System Interface Definitions volume of  
38388 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

38389 The following options shall be supported:

38390 **-c** Do not copy local file to the spool directory for transfer to the remote machine  
38391 (default).38392 UN **-C** Force the copy of local files to the spool directory for transfer.38393 **-d** Make all necessary directories for the file copy (default).38394 UN **-f** Do not make intermediate directories for the file copy.38395 UN **-j** Write the job identification string to standard output. This job identification can be  
38396 used by *uustat* to obtain the status or terminate a job.38397 **-m** Send mail to the requester when the copy is completed.38398 UN **-n user** Notify *user* on the remote system that a file was sent.38399 UN **-r** Do not start the file transfer; just queue the job.

## 38400 OPERANDS

38401 The following operands shall be supported:

38402 *destination-file, source-file*38403 A path name of a file to be copied to, or from, respectively. Either name can be a  
38404 path name on the local machine, or can have the form:38405 *system-name!pathname*38406 where *system-name* is taken from a list of system names that *uucp* knows about.  
38407 The destination *system-name* can also be a list of names such as:38408 *system-name!system-name!...!system-name!pathname*38409 in which case, an attempt is made to send the file via the specified route to the  
38410 destination. Care should be taken to ensure that intermediate nodes in the route  
38411 are willing to forward information.

38412 The shell pattern matching notation characters '?', '\*', and "[...]" appearing  
38413 in *pathname* are expanded on the appropriate system.

38414 Path names can be one of:

- 38415 1. An absolute path name.
- 38416 2. A path name preceded by *~user* where *user* is a login name on the specified  
38417 system and is replaced by that user's login directory. Note that if an invalid  
38418 login is specified, the default is to the public directory (called *PUBDIR*; the  
38419 actual location of *PUBDIR* is implementation-dependent).
- 38420 3. A path name preceded by *~/destination* where *destination* is appended to  
38421 *PUBDIR*.

38422 **Note:** This destination is treated as a file name unless more than one file  
38423 is being transferred by this request or the destination is already a  
38424 directory. To ensure that it is a directory, follow the destination  
38425 with a '/'. For example, *~/dan/* as the destination makes the  
38426 directory **PUBDIR/dan** if it does not exist and put the requested  
38427 files in that directory.

- 38428 4. Anything else is prefixed by the current directory.

38429 If the result is an erroneous path name for the remote system, the copy fails. If the  
38430 *destination-file* is a directory, the last part of the *source-file* name is used.

38431 The read, write, and execute permissions given by *uucp* are implementation-  
38432 dependent.

#### 38433 **STDIN**

38434 Not used.

#### 38435 **INPUT FILES**

38436 The files to be copied are regular files.

#### 38437 **ENVIRONMENT VARIABLES**

38438 The following environment variables shall affect the execution of *uucp*:

38439 **LANG** Provide a default value for the internationalization variables that are unset or null.  
38440 If *LANG* is unset or null, the corresponding value from the implementation-  
38441 dependent default locale shall be used. If any of the internationalization variables  
38442 contains an invalid setting, the utility shall behave as if none of the variables had  
38443 been defined.

38444 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
38445 internationalization variables.

38446 **LC\_COLLATE**  
38447 Determine the locale for the behavior of ranges, equivalence classes and multi-  
38448 character collating elements within bracketed file name patterns.

38449 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
38450 characters (for example, single-byte as opposed to multi-byte characters in  
38451 arguments and input files) and the behavior of character classes within bracketed  
38452 file name patterns (for example, "[[:lower:]]\*").

#### 38453 **LC\_MESSAGES**

38454 Determine the locale that should be used to affect the format and contents of  
38455 diagnostic messages written to standard error, and informative messages written

38456 to standard output.

38457 *LC\_TIME* Determine the format of date and time strings output by *uucp*.

38458 *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

38459 *TZ* Determine the timezone used with date and time strings.

#### 38460 **ASYNCHRONOUS EVENTS**

38461 Default.

#### 38462 **STDOUT**

38463 Not used.

#### 38464 **STDERR**

38465 Used only for diagnostic messages.

#### 38466 **OUTPUT FILES**

38467 The output files (which may be on other systems) are copies of the input files.

38468 If the *-m* is used, mail files are modified.

#### 38469 **EXTENDED DESCRIPTION**

38470 None.

#### 38471 **EXIT STATUS**

38472 The following exit values shall be returned:

38473 0 Successful completion.

38474 >0 An error occurred.

#### 38475 **CONSEQUENCES OF ERRORS**

38476 Default.

#### 38477 **APPLICATION USAGE**

38478 The domain of remotely accessible files can (and for obvious security reasons usually should) be  
38479 severely restricted.

38480 Note that the '!' character in addresses has to be escaped when using *cs*h as a command  
38481 interpreter because of its history substitution syntax. For *ksh* and *sh* the escape is not necessary,  
38482 but may be used.

38483 Typical implementations of this utility require a communications line configured to use the  
38484 System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal  
38485 Interface, but other communications means may be used. On systems where there are no  
38486 available communications means (either temporarily or permanently), this utility shall write an  
38487 error message describing the problem and exit with a non-zero exit status.

38488 As noted above, shell metacharacters appearing in path names are expanded on the appropriate  
38489 system. On an internationalized system, this is done under the control of local settings of  
38490 *LC\_COLLATE* and *LC\_CTYPE*. Thus, care should be taken when using bracketed file name  
38491 patterns, as collation and typing rules may vary from one system to another. Also be aware that  
38492 certain types of expression (that is, equivalence classes, character classes, and collating symbols)  
38493 need not be supported on non-internationalized systems.

#### 38494 **EXAMPLES**

38495 None.

38496 **RATIONALE**

38497 None.

38498 **FUTURE DIRECTIONS**

38499 None.

38500 **SEE ALSO**38501 *mailx, uuencode, uustat, uux*38502 **CHANGE HISTORY**

38503 First released in Issue 2.

38504 **Issue 4**

38505 Format reorganized.

38506 Split into a separate description.

38507 Utility Syntax Guidelines support mandated.

38508 Internationalized environment variable support mandated.

38509 Presence of the utility mandated, even on systems where no communications are available.

38510 **NAME**

38511           uudecode — decode a binary file

38512 **SYNOPSIS**38513 UP       uudecode [-o *outfile*][*file*]

38514

38515 **DESCRIPTION**

38516       The *uudecode* utility shall read a file, or standard input if no file is specified, that includes data  
 38517       created by the *uuencode* utility. The *uudecode* utility shall scan the input file, searching for data  
 38518       compatible with one of the formats specified in *uuencode* and attempt to create or overwrite the  
 38519       file described by the data (or overridden by the *-o* option). The path name shall be contained in  
 38520       the data or specified by the *-o* option. The file access permission bits and contents for the file to  
 38521       be produced shall be contained in that data. The mode bits of the created file (other than  
 38522       standard output) shall be set from the file access permission bits contained in the data; that is,  
 38523       other attributes of the mode, including the file mode creation mask (see *umask*), shall not affect  
 38524       the file being produced.

38525       If the path name of the file to be produced exists, and the user does not have write permission on  
 38526       that file, *uudecode* shall terminate with an error. If the path name of the file to be produced exists,  
 38527       and the user has write permission on that file, the existing file shall be overwritten.

38528       If the input data was produced by *uuencode* on a system with a different number of bits per byte  
 38529       than on the target system, the results of *uudecode* are unspecified.

38530 **OPTIONS**

38531       The *uudecode* utility shall conform to the System Interface Definitions volume of  
 38532       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

38533       The following option shall be supported by the implementation:

38534       *-o outfile*   A path name of a file that shall be used instead of any path name contained in the  
 38535       input data. Specifying an *outfile* option-argument of */dev/stdout* shall indicate  
 38536       standard output.

38537 **OPERANDS**

38538       The following operand shall be supported:

38539       *file*           The path name of a file containing the output of *uuencode*.

38540 **STDIN**

38541       See the INPUT FILES section.

38542 **INPUT FILES**

38543       The input files shall be files containing the output of *uuencode*.

38544 **ENVIRONMENT VARIABLES**

38545       The following environment variables shall affect the execution of *uudecode*:

38546       *LANG*           Provide a default value for the internationalization variables that are unset or null.  
 38547       If *LANG* is unset or null, the corresponding value from the implementation-  
 38548       dependent default locale shall be used. If any of the internationalization variables  
 38549       contains an invalid setting, the utility shall behave as if none of the variables had  
 38550       been defined.

38551       *LC\_ALL*          If set to a non-empty string value, override the values of all the other  
 38552       internationalization variables.

38553       *LC\_CTYPE*       Determine the locale for the interpretation of sequences of bytes of text data as  
 38554       characters (for example, single-byte as opposed to multi-byte characters in



38555 arguments and input files).

38556 **LC\_MESSAGES**

38557 Determine the locale that should be used to affect the format and contents of

38558 diagnostic messages written to standard error.

38559 XSI **NLSPATH** Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

38560 **ASYNCHRONOUS EVENTS**

38561 Default.

38562 **STDOUT**

38563 If the file data header encoded by *uuencode* is **-** or **/dev/stdout**, or the **-o /dev/stdout** option

38564 overrides the file data, the standard output shall be in the same format as the file originally

38565 encoded by *uuencode*. Otherwise, the standard output shall not be used.

38566 **STDERR**

38567 Used only for diagnostic messages.

38568 **OUTPUT FILES**

38569 The output file shall be in the same format as the file originally encoded by *uuencode*.

38570 **EXTENDED DESCRIPTION**

38571 None.

38572 **EXIT STATUS**

38573 The following exit values shall be returned:

38574 0 Successful completion.

38575 >0 An error occurred.

38576 **CONSEQUENCES OF ERRORS**

38577 Default.

38578 **APPLICATION USAGE**

38579 The user who is invoking *uudecode* must have write permission on any file being created.

38580 The output of *uuencode* is essentially an encoded bit stream that is not cognizant of byte

38581 boundaries. It is possible that a 9-bit byte target machine can process input from an 8-bit source,

38582 if it is aware of the requirement, but the reverse is unlikely to be satisfying. Of course, the only

38583 data that is meaningful for such a transfer between architectures is generally character data.

38584 Application writers should note that this utility need not be provided on systems that do not

38585 support the User Portability Utilities option.

38586 **EXAMPLES**

38587 None.

38588 **RATIONALE**

38589 Input files are not necessarily text files, as stated by an early proposal. Although the *uuencode*

38590 output is a text file, that output could have been wrapped within another file or mail message

38591 that is not a text file.

38592 The **-o** option is not historical practice, but was added at the request of WG15 so that the user

38593 could override the target path name without having to edit the input data itself.

38594 In early drafts, the **[-o outfile]** option-argument allowed the use of **-** to mean standard output.

38595 The symbol **-** has only been used previously in IEEE Std. 1003.1-200x as a standard input

38596 indicator. The developers of the standard did not wish to overload the meaning of **-** in this

38597 manner. The **/dev/stdout** concept exists on most modern systems. The **/dev/stdout** syntax does

38598 not refer to a new special file. It is just a magic cookie to specify standard output. |

38599 **FUTURE DIRECTIONS**

38600 None.

38601 **SEE ALSO**

38602 *uuencode*

38603 **CHANGE HISTORY**

38604 First released in Issue 4.

38605 **Issue 6**

38606 This utility is now marked as part of the User Portability Utilities option. |

38607 The **-o outfile** option is added, as specified in the IEEE P1003.2b draft standard. |

38608 The normative text is reworded to avoid use of the term “must” for application requirements. |

38609 **NAME**

38610 uuencode — encode a binary file

38611 **SYNOPSIS**38612 UP uuencode [-m][*file*] *decode\_pathname*

38613

38614 **DESCRIPTION**

38615 The *uuencode* utility shall write an encoded version of the named input file, or standard input if  
 38616 no *file* is specified, to standard output. The output shall be encoded using one of the algorithms  
 38617 described in the STDOUT section and shall include the file access permission bits (in *chmod* octal  
 38618 or symbolic notation) of the input file and the *decode\_pathname*, for re-creation of the file on  
 38619 another system that conforms to this volume of IEEE Std. 1003.1-200x.

38620 **OPTIONS**

38621 The *uuencode* utility shall conform to the System Interface Definitions volume of  
 38622 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

38623 The following option shall be supported by the implementation:

38624 **-m** Encode the output using the MIME Base64 algorithm described below. If **-m** is not  
 38625 specified, the historical algorithm described in STDOUT shall be used.

38626 **OPERANDS**

38627 The following operands shall be supported:

38628 *decode\_pathname*

38629 The path name of the file into which the *uudecode* utility shall place the decoded  
 38630 file. Specifying a *decode\_pathname* operand of **/dev/stdout** shall indicate that  
 38631 *uudecode* is to use standard output. If there are characters in *decode\_pathname* that  
 38632 are not in the portable file name character set the results are unspecified.

38633 *file* A path name of the file to be encoded.

38634 **STDIN**

38635 See the INPUT FILES section.

38636 **INPUT FILES**

38637 Input files can be files of any type.

38638 **ENVIRONMENT VARIABLES**

38639 The following environment variables shall affect the execution of *uuencode*:

38640 **LANG** Provide a default value for the internationalization variables that are unset or null.  
 38641 If **LANG** is unset or null, the corresponding value from the implementation-  
 38642 dependent default locale shall be used. If any of the internationalization variables  
 38643 contains an invalid setting, the utility shall behave as if none of the variables had  
 38644 been defined.

38645 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 38646 internationalization variables.

38647 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 38648 characters (for example, single-byte as opposed to multi-byte characters in  
 38649 arguments and input files).

38650 **LC\_MESSAGES**

38651 Determine the locale that should be used to affect the format and contents of  
 38652 diagnostic messages written to standard error.

38653 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

38654 **ASYNCHRONOUS EVENTS**

38655 Default.

38656 **STDOUT**

38657 **uuencode Base64 Algorithm**

38658 The standard output shall be a text file (encoded in the character set of the current locale) that  
38659 begins with the line:

38660 "begin-base64ss\n", *<mode>*, *decode\_pathname*

38661 and ends with the line:

38662 "====\n"

38663 In both cases, the lines shall have no preceding or trailing <blank>s.

38664 The encoding process represents 24-bit groups of input bits as output strings of four encoded  
38665 characters. Preceding from left to right, a 24-bit input group shall be formed by concatenating  
38666 three 8-bit input groups. These 24-bit then shall be treated as four concatenated 6-bit groups,  
38667 each of which shall be translated into a single digit in the base64 alphabet. When encoding a bit  
38668 stream via the base64 encoding, the bit stream shall be presumed to be ordered with the most-  
38669 significant bit first. That is, the first bit in the stream shall be the high-order bit in the first byte,  
38670 and the eighth bit shall be the low-order bit in the first byte, and so on. Each 6-bit group is used  
38671 as an index into an array of 64 printable characters, as shown in Table 4-21.

38672 **Table 4-21 uuencode Base64 Values**

| Value | Encoding | Value | Encoding | Value | Encoding | Value | Encoding |
|-------|----------|-------|----------|-------|----------|-------|----------|
| 0     | A        | 17    | R        | 34    | i        | 51    | z        |
| 1     | B        | 18    | S        | 35    | j        | 52    | 0        |
| 2     | C        | 19    | T        | 36    | k        | 53    | 1        |
| 3     | D        | 20    | U        | 37    | l        | 54    | 2        |
| 4     | E        | 21    | V        | 38    | m        | 55    | 3        |
| 5     | F        | 22    | W        | 39    | n        | 56    | 4        |
| 6     | G        | 23    | X        | 40    | o        | 57    | 5        |
| 7     | H        | 24    | Y        | 41    | p        | 58    | 6        |
| 8     | I        | 25    | Z        | 42    | q        | 59    | 7        |
| 9     | J        | 26    | a        | 43    | r        | 60    | 8        |
| 10    | K        | 27    | b        | 44    | s        | 61    | 9        |
| 11    | L        | 28    | c        | 45    | t        | 62    | +        |
| 12    | M        | 29    | d        | 46    | u        | 63    | /        |
| 13    | N        | 30    | e        | 47    | v        | (pad) | =        |
| 14    | O        | 31    | f        | 48    | w        |       |          |
| 15    | P        | 32    | g        | 49    | x        |       |          |
| 16    | Q        | 33    | h        | 50    | y        |       |          |

38691 The character referenced by the index shall be placed in the output string.

38692 The output stream (encoded bytes) shall be represented in lines of no more than 76 characters  
38693 each. All line breaks or other characters not found in the table shall be ignored by decoding  
38694 software (see *uudecode*).

38695 Special processing shall be performed if fewer than 24 bits are available at the end of a message  
38696 or encapsulated part of a message. A full encoding quantum shall always be completed at the

38697 end of a message. When fewer than 24 input bits are available in an input group, zero bits shall  
 38698 be added (on the right) to form an integral number of 6-bit groups. Output character positions  
 38699 that are not required to represent actual input data shall be set to the character '='. Since all  
 38700 base64 input is an integral number of octets, only the following cases can arise:

- 38701 1. The final quantum of encoding input is an integral multiple of 24 bits; here, the final unit of  
 38702 encoded output shall be an integral multiple of 4 characters with no '=' padding.
- 38703 2. The final quantum of encoding input is exactly 8 bits; here, the final unit of encoded output  
 38704 shall be two characters followed by two '=' padding characters.
- 38705 3. The final quantum of encoding input is exactly 16 bits; here, the final unit of encoded  
 38706 output shall be three characters followed by one '=' padding character.
- 38707 4. The terminating "====" evaluates to nothing and denotes the end of the encoded data.

### 38708 uuencode Historical Algorithm

38709 The standard output shall be a text file (encoded in the character set of the current locale) that  
 38710 begins with the line:

38711 "beginΔ%sΔ%s\n" <mode>, <decode\_pathname>

38712 and ends with the line:

38713 end\n

38714 In both cases, the lines shall have no preceding or trailing <blank> characters.

38715 The algorithm that shall be used for lines in between **begin** and **end** takes three octets as input  
 38716 and writes four characters of output by splitting the input at six-bit intervals into four octets,  
 38717 containing data in the lower six bits only. These octets shall be converted to characters by adding  
 38718 a value of 0x20 to each octet, so that each octet is in the range 0x20-0x5f, and then it shall be  
 38719 assumed to represent a printable character in the ISO/IEC 646:1991 standard encoded character  
 38720 set. It then shall be translated into the corresponding character codes for the codeset in use in the  
 38721 current locale. (For example, the octet 0x41, representing 'A', would be translated to 'A' in the  
 38722 current codeset, such as 0xc1 if it were EBCDIC.)

38723 Where the bits of two octets are combined, the least significant bits of the first octet shall be  
 38724 shifted left and combined with the most significant bits of the second octet shifted right. Thus  
 38725 the three octets *A*, *B*, *C* shall be converted into the four octets:

38726  $0x20 + ((A \gg 2) \& 0x3F)$   
 38727  $0x20 + (((A \ll 4) \mid ((B \gg 4) \& 0xF)) \& 0x3F)$   
 38728  $0x20 + (((B \ll 2) \mid ((C \gg 6) \& 0x3)) \& 0x3F)$   
 38729  $0x20 + ((C) \& 0x3F)$

38730 These octets then shall be translated into the local character set.

38731 Each encoded line contains a length character, equal to the number of characters to be decoded  
 38732 plus 0x20 translated to the local character set as described above, followed by the encoded  
 38733 characters. The maximum number of octets to be encoded on each line shall be 45.

### 38734 STDERR

38735 Used only for diagnostic messages.

### 38736 OUTPUT FILES

38737 None.

38738 **EXTENDED DESCRIPTION**

38739 None.

38740 **EXIT STATUS**

38741 The following exit values shall be returned:

38742 0 Successful completion.

38743 &gt;0 An error occurred.

38744 **CONSEQUENCES OF ERRORS**

38745 Default.

38746 **APPLICATION USAGE**

38747 The file is expanded by 35 percent (each three octets become four, plus control information)  
38748 causing it to take longer to transmit.

38749 Since this utility is intended to create files to be used for data interchange between systems with  
38750 possibly different codesets, and to represent binary data as a text file, the ISO/IEC 646:1991  
38751 standard was chosen for a midpoint in the algorithm as a known reference point. The output  
38752 from *uuencode* is a text file on the local system. If the output were in the ISO/IEC 646:1991  
38753 standard codeset, it might not be a text file (at least because the <newline> characters might not  
38754 match), and the goal of creating a text file would be defeated. If this text file was then carried to  
38755 another machine with the same codeset, it would be perfectly compatible with that system's  
38756 *uudecode*. If it was transmitted over a mail system or sent to a machine with a different codeset,  
38757 it is assumed that, as for every other text file, some translation mechanism would convert it (by  
38758 the time it reached a user on the other system) into an appropriate codeset. This translation only  
38759 makes sense from the local codeset, not if the file has been put into a ISO/IEC 646:1991 standard  
38760 representation first. Similarly, files processed by *uuencode* can be placed in *pax* archives,  
38761 intermixed with other text files in the same codeset.

38762 The algorithm is described in terms of 8-bit quantities, or octets. Since no byte alignment is  
38763 implied, it encodes data from machines with any number of bits per byte. However, unless that  
38764 encoded data is then decoded on a machine with the same number of bits per byte, the output  
38765 might not be useful.

38766 Application writers should note that this utility need not be provided on systems that do not  
38767 support the User Portability Utilities option.

38768 **EXAMPLES**

38769 None.

38770 **RATIONALE**

38771 A new algorithm was added at the request of the international community to parallel work in  
38772 RFC 2045 (MIME). As with the historical *uuencode* format, the Base64 Content-Transfer-Encoding  
38773 is designed to represent arbitrary sequences of octets in a form that is not humanly readable. A  
38774 65-character subset of the ISO/IEC 646:1991 standard is used, enabling 6 bits to be represented  
38775 per printable character. (The extra 65th character, '=', is used to signify a special processing  
38776 function.)

38777 This subset has the important property that it is represented identically in all versions of the  
38778 ISO/IEC 646:1991 standard, including US ASCII, and all characters in the subset are also  
38779 represented identically in all versions of EBCDIC. The historical *uuencode* algorithm does not  
38780 share this property, which is the reason that a second algorithm was added to the ISO POSIX-2  
38781 standard.

38782 The string "====" was used for the termination instead of the end used in the original format  
38783 because the latter is a string that could be valid encoded input.

38784 In an early draft, the **-m** option was named **-b** (for Base64), but it was renamed to reflect its  
38785 relationship to the RFC 2045. A **-u** was also present to invoke the default algorithm, but since  
38786 this was not historical practice, it was omitted as being unnecessary.

38787 See the RATIONALE section in *uudecode* for the derivation of the **/dev/stdout** symbol.

38788 **FUTURE DIRECTIONS**

38789 None.

38790 **SEE ALSO**

38791 *mailx*, *uudecode*

38792 **CHANGE HISTORY**

38793 First released in Issue 4.

38794 **Issue 6**

38795 This utility is now marked as part of the User Portability Utilities option.

38796 The Base64 algorithm and the ability to output to **/dev/stdout** are added as specified in the  
38797 IEEE P1003.2b draft standard.

38798 **NAME**

38799           uustat — uucp status inquiry and job control

38800 **SYNOPSIS**38801 UN XSI    uustat [ -q | -k *jobid* | -r *jobid* ]38802 XSI       uustat [-s *system*][-u *user*]38803 **DESCRIPTION**38804           The *uustat* utility shall display the status of, or cancel, previously specified *uucp* requests, or  
38805           provide general status on *uucp* connections to other systems.38806           When no options are given, *uustat* shall write to standard output the status of all *uucp* requests  
38807           issued by the current user.38808           Typical implementations of this utility require a communications line configured to use the  
38809           System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal  
38810           Interface, but other communications means may be used. On systems where there are no  
38811           available communications means (either temporarily or permanently), this utility shall write an  
38812           error message describing the problem and exits with a non-zero exit status.38813 **OPTIONS**38814           The *uustat* utility shall conform to the System Interface Definitions volume of  
38815           IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

38816           The following options shall be supported:

38817 UN       **-q**       Write the jobs queued for each machine.38818       **-k *jobid***   Kill the *uucp* request whose job identification is *jobid*. The application shall ensure  
38819                      that the killed *uucp* request belongs to the person invoking *uustat* unless that user  
38820                      has appropriate privileges.38821       **-r *jobid***   Rejuvenate *jobid*. The files associated with *jobid* are touched so that their  
38822                      modification time is set to the current time. This prevents the cleanup program  
38823                      from deleting the job until the jobs modification time reaches the limit imposed by  
38824                      the program.38825       **-s *system***   Write the status of all *uucp* requests for remote system *system*.38826       **-u *user***     Write the status of all *uucp* requests issued by *user*.38827 **OPERANDS**

38828           None.

38829 **STDIN**

38830           Not used.

38831 **INPUT FILES**

38832           None.

38833 **ENVIRONMENT VARIABLES**38834           The following environment variables shall affect the execution of *uustat*:38835       **LANG**       Provide a default value for the internationalization variables that are unset or null.  
38836                      If *LANG* is unset or null, the corresponding value from the implementation-  
38837                      dependent default locale shall be used. If any of the internationalization variables  
38838                      contains an invalid setting, the utility shall behave as if none of the variables had  
38839                      been defined.



38840 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 38841 internationalization variables.

38842 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 38843 characters (for example, single-byte as opposed to multi-byte characters in  
 38844 arguments).

38845 *LC\_MESSAGES*  
 38846 Determine the locale that should be used to affect the format and contents of  
 38847 diagnostic messages written to standard error, and informative messages written  
 38848 to standard output.

38849 *LC\_TIME* Determine the format of date and time strings output by *uustat*.

38850 *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

38851 *TZ* Determine the timezone used with date and time strings.

38852 **ASYNCHRONOUS EVENTS**  
 38853 Default.

38854 **STDOUT**  
 38855 The standard output shall consist of information about each job selected, in an unspecified  
 38856 format. The information shall include at least the job ID, the user ID or name, and the remote  
 38857 system name.

38858 **STDERR**  
 38859 Used only for diagnostic messages.

38860 **OUTPUT FILES**  
 38861 None.

38862 **EXTENDED DESCRIPTION**  
 38863 None.

38864 **EXIT STATUS**  
 38865 The following exit values shall be returned:  
 38866 0 Successful completion.  
 38867 >0 An error occurred.

38868 **CONSEQUENCES OF ERRORS**  
 38869 Default.

38870 **APPLICATION USAGE**  
 38871 None.

38872 **EXAMPLES**  
 38873 None.

38874 **RATIONALE**  
 38875 None.

38876 **FUTURE DIRECTIONS**  
 38877 None.

38878 **SEE ALSO**  
 38879 *uucp*

38880 **CHANGE HISTORY**

38881 First released in Issue 2.

38882 **Issue 4**

38883 Format reorganized.

38884 Utility Syntax Guidelines support mandated.

38885 Internationalized environment variable support mandated.

38886 Presence of the utility mandated, even on systems where no communications are available. |

38887 **Issue 6** |

38888 The normative text is reworded to avoid use of the term “must” for application requirements. |

38889 **NAME**38890 **uux** — remote command execution38891 **SYNOPSIS**38892 XSI `uux [-np] command-string`38893 UN XSI `uux [-jnp] command-string`38894 **DESCRIPTION**

38895 The *uux* utility shall gather zero or more files from various systems, execute a shell pipeline (see  
 38896 Section 2.9 on page 67) on a specified system, and then send the standard output of the  
 38897 command to a file on a specified system. Only the first command of a pipeline can have a  
 38898 *system-name!* prefix. All other commands in the pipeline shall be executed on the system of the  
 38899 first command.

38900 The following restrictions are applicable to the shell pipeline processed by *uux*:

- 38901 • In gathering files from different systems, path name expansion is not performed by *uux*.  
 38902 Thus, a request such as:

38903 `uux "c89 remsys!~/*.c"`

38904 would attempt to copy the file named literally *\*.c* to the local system.

- 38905 • The redirection operators "*>>*", "*<<*", "*>|*", and "*>&*" cannot be used.
- 38906 • The reserved word *!* cannot be used at the head of the pipeline to modify the exit status.
- 38907 • Alias substitution is not performed.

38908 A file name can be specified as for *uucp*; it can be an absolute path name, a path name preceded  
 38909 by *~name* (which is replaced by the corresponding login directory), a path name specified as  
 38910 *~/dest* (*dest* is prefixed by the public directory called *PUBDIR*; the actual location of *PUBDIR* is  
 38911 implementation-dependent), or a simple file name (which is prefixed by *uux* with the current  
 38912 directory). See *uucp* on page 1008 for the details.

38913 The execution of commands on remote systems shall take place in an execution directory known  
 38914 to the *uucp* system. All files required for the execution shall be put into this directory unless they  
 38915 already reside on that machine. Therefore, the application shall ensure that non-local file names  
 38916 (without path or machine reference) are unique within the *uux* request.

38917 The *uux* utility shall attempt to get all files to the execution system. For files that are output files,  
 38918 the application shall ensure that the file name is escaped using parentheses.

38919 The remote system shall notify the user by mail if the requested command on the remote system  
 38920 was disallowed or the files were not accessible. This notification can be turned off by the *-n*  
 38921 option.

38922 Typical implementations of this utility require a communications line configured to use the  
 38923 System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11, General Terminal  
 38924 Interface, but other communications means may be used. On systems where there are no  
 38925 available communications means (either temporarily or permanently), this utility shall write an  
 38926 error message describing the problem and exits with a non-zero exit status.

38927 The *uux* utility cannot guarantee support for all character encodings in all circumstances. For  
 38928 example, transmission data may be restricted to 7 bits by the underlying network, 8-bit data and  
 38929 file names need not be portable to non-internationalized systems, and so on. Under these  
 38930 circumstances, it is recommended that only characters defined in the ISO/IEC 646:1991  
 38931 standard International Reference Version (equivalent to ASCII) 7-bit range of characters be used  
 38932 and that only characters defined in the Portable File Name Character Set be used for naming

38933 files.

### 38934 OPTIONS

38935 The *uux* utility shall conform to the System Interface Definitions volume of  
38936 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

38937 The following options shall be supported:

38938 **-p** Make the standard input to *uux* the standard input to the *command-string*.

38939 UN **-j** Write the job identification string to standard output. This job identification can be  
38940 used by *uustat* to obtain the status or terminate a job.

38941 **-n** Do not notify the user if the command fails.

### 38942 OPERANDS

38943 The following operand shall be supported:

38944 *command-string*

38945 A string made up of one or more arguments that are similar to normal command  
38946 arguments, except that the command and any file names can be prefixed by  
38947 *system-name!*. A null *system-name* shall be interpreted as the local system.

### 38948 STDIN

38949 The standard input shall not be used unless the *'-'* or **-p** option is specified; in those cases, the  
38950 standard input shall be made the standard input of the *command-string*.

### 38951 INPUT FILES

38952 Input files shall be selected according to the contents of *command-string*.

### 38953 ENVIRONMENT VARIABLES

38954 The following environment variables shall affect the execution of *uux*:

38955 **LANG** Provide a default value for the internationalization variables that are unset or null.  
38956 If **LANG** is unset or null, the corresponding value from the implementation-  
38957 dependent default locale shall be used. If any of the internationalization variables  
38958 contains an invalid setting, the utility shall behave as if none of the variables had  
38959 been defined.

38960 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
38961 internationalization variables.

38962 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
38963 characters (for example, single-byte as opposed to multi-byte characters in  
38964 arguments).

38965 **LC\_MESSAGES**  
38966 Determine the locale that should be used to affect the format and contents of  
38967 diagnostic messages written to standard error.

38968 **NLSPATH** Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

### 38969 ASYNCHRONOUS EVENTS

38970 Default.

### 38971 STDOUT

38972 The standard output shall be not used unless the **-j** option is specified; in that case, the job  
38973 identification string shall be written to standard output in the following format:

38974 "%s\n", <jobid>

38975 **STDERR**

38976           Used only for diagnostic messages.

38977 **OUTPUT FILES**

38978           Output files shall be created or written, or both, according to the contents of *command-string*.

38979           If the **-n** is not used, mail files shall be modified following any command or file-access failures  
38980           on the remote system.

38981 **EXTENDED DESCRIPTION**

38982           None.

38983 **EXIT STATUS**

38984           The following exit values shall be returned:

38985           0   Successful completion.

38986           >0  An error occurred.

38987 **CONSEQUENCES OF ERRORS**

38988           Default.

38989 **APPLICATION USAGE**

38990           Note that, for security reasons, many installations limit the list of commands executable on  
38991           behalf of an incoming request from *uux*. Many sites permit little more than the receipt of mail  
38992           via *uux*.

38993           Any characters special to the command interpreter should be quoted either by quoting the entire  
38994           *command-string* or quoting the special characters as individual arguments.

38995           As noted in *uucp*, shell pattern matching notation characters appearing in path names are  
38996           expanded on the appropriate local system. This is done under the control of local settings of  
38997           *LC\_COLLATE* and *LC\_CTYPE*. Thus, care should be taken when using bracketed file name  
38998           patterns, as collation and typing rules may vary from one system to another. Also be aware that  
38999           certain types of expression (that is, equivalence classes, character classes, and collating symbols)  
39000           need not be supported on non-internationalized systems.

39001 **EXAMPLES**

39002           1. The following command gets **file1** from system **a** and **file2** file from system **b**, executes *diff*  
39003           on the local system, and puts the results in **file.diff** in the local *PUBDIR* directory.  
39004           (*PUBDIR* is the *uucp* public directory on the local system.)

39005           uux "!diff a!/usr/file1 b!/a4/file2 >!~/file.diff"

39006           2. The following command fails because *uux* places all files copied to a system in the same  
39007           working directory. Although the files **xyz** are from two different systems, their file names  
39008           are the same and conflict.

39009           uux "!diff a!/usr1/xyz b!/usr2/xyz >!~/xyz.diff"

39010           3. The following command succeeds (assuming *diff* is permitted on system **a**) because the file  
39011           local to system **a** is not copied to the working directory, and hence does not conflict the file  
39012           from system **c**.

39013           uux "a!diff a!/usr/xyz c!/usr/xyz >!~/xyz.diff"

39014 **RATIONALE**

39015 None.

39016 **FUTURE DIRECTIONS**

39017 A version of *uux* that fully supports the System Interface Definitions volume of  
39018 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines may be introduced in a future  
39019 issue.

39020 **SEE ALSO**39021 *uucp, uuencode, uustat*39022 **CHANGE HISTORY**

39023 First released in Issue 2.

39024 **Issue 4**

39025 Format reorganized.

39026 Exceptions to Utility Syntax Guidelines conformance noted.

39027 Internationalized environment variable support mandated.

39028 Presence of the utility mandated, even on systems where no communications are available.

39029 **Issue 6**

39030 The obsolescent SYNOPSIS is removed.

39031 The normative text is reworded to avoid use of the term “must” for application requirements.

39032 **NAME**39033 **val** — validate SCCS files (**DEVELOPMENT**)39034 **SYNOPSIS**39035 **XSI** **val** -39036 **val** [-s][-m *name*][-r *SID*][-y *type*] *file*...

39037

39038 **DESCRIPTION**

39039 The *val* utility shall determine whether the specified *file* is an SCCS file meeting the  
 39040 characteristics specified by the options.

39041 **OPTIONS**

39042 The *val* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 39043 Section 12.2, Utility Syntax Guidelines, except that the usage of the '-' operand is not strictly as  
 39044 intended by the guidelines (that is, reading options and operands from standard input).

39045 The following options shall be supported:

39046 **-m *name*** Specify a *name*, which is compared with the SCCS %M% keyword in *file*; see *get* on  
 39047 page 510.

39048 **-r *SID*** Specify a *SID* (SCCS Identification String), an SCCS delta number. A check is made  
 39049 to determine if the *SID* is ambiguous (for example, **-r 1** is ambiguous because it  
 39050 physically does not exist but implies 1.1, 1.2, and so on, which may exist) or invalid  
 39051 (for example, **-r 1.0** or **-r 1.1.0** are invalid because neither case can exist as a valid  
 39052 delta number). If the *SID* is valid and not ambiguous, a check is made to  
 39053 determine whether it actually exists.

39054 **-s** Silence the diagnostic message normally written to standard output for any error  
 39055 that is detected while processing each named file on a given command line.

39056 **-y *type*** Specify a *type*, which is compared with the SCCS %Y% keyword in *file*; see *get* on  
 39057 page 510.

39058 **OPERANDS**

39059 The following operands shall be supported:

39060 ***file*** A path name of an existing SCCS file. If a single instance *file* is specified as '-',  
 39061 and if no options are specified, the standard input shall be read: each line is  
 39062 independently processed as if it were a command line argument list. (However,  
 39063 the line is not subjected to any of the shell word expansions, such as parameter  
 39064 expansion or quote removal.)

39065 **STDIN**

39066 The standard input is a text file used only when the *file* operand is specified as '-'.

39067 **INPUT FILES**

39068 Any SCCS files processed are files of an unspecified format.

39069 **ENVIRONMENT VARIABLES**

39070 The following environment variables shall affect the execution of *val*:

39071 **LANG** Provide a default value for the internationalization variables that are unset or null.  
 39072 If *LANG* is unset or null, the corresponding value from the implementation-  
 39073 dependent default locale shall be used. If any of the internationalization variables  
 39074 contains an invalid setting, the utility shall behave as if none of the variables had  
 39075 been defined.

39076        *LC\_ALL*     If set to a non-empty string value, override the values of all the other  
 39077                   internationalization variables.

39078        *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 39079                   characters (for example, single-byte as opposed to multi-byte characters in  
 39080                   arguments and input files).

39081        *LC\_MESSAGES*  
 39082                   Determine the locale that should be used to affect the format and contents of  
 39083                   diagnostic messages written to standard error, and informative messages written  
 39084                   to standard output.

39085        *NLSPATH*   Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

39086 **ASYNCHRONOUS EVENTS**  
 39087        Default.

39088 **STDOUT**  
 39089        The standard output shall consist of informative messages about either:

39090            1. Each file processed

39091            2. Each command line read from standard input

39092        If the standard input is not used, for each *file* operand yielding a discrepancy, the output line  
 39093        shall have the following format:

39094        "%s: %s\n", <pathname>, <unspecified string>

39095        If standard input is used, a line of input shall be written before each of the preceding lines for  
 39096        files containing discrepancies:

39097        "%s:\n", <input line>

39098 **STDERR**  
 39099        Not used.

39100 **OUTPUT FILES**  
 39101        None.

39102 **EXTENDED DESCRIPTION**  
 39103        None.

39104 **EXIT STATUS**  
 39105        The 8-bit code returned by *val* is a disjunction of the possible errors, that is, it can be interpreted  
 39106        as a bit string where set bits are interpreted as follows:

39107        0x80    =   Missing file argument.

39108        0x40    =   Unknown or duplicate option.

39109        0x20    =   Corrupted SCCS file.

39110        0x10    =   Cannot open file or file not SCCS.

39111        0x08    =   *SID* is invalid or ambiguous.

39112        0x04    =   *SID* does not exist.

39113        0x02    =   %Y%, -y mismatch.

39114        0x01    =   %M%, -m mismatch.

39115        Note that *val* can process two or more files on a given command line and can process multiple  
 39116        command lines (when reading the standard input). In these cases an aggregate code shall be  
 39117        returned: a logical OR of the codes generated for each command line and file processed.



**39118 CONSEQUENCES OF ERRORS**

39119 Default.

**39120 APPLICATION USAGE**

39121 Since the *val* exit status sets the 0x80 bit, shell applications checking "\$?" cannot tell if it  
39122 terminated due to a missing file argument or receipt of a signal.

**39123 EXAMPLES**

39124 In a directory with three SCCS files, **s.x** (of **t** type "text"), **s.y**, and **s.z** (a corrupted file), the  
39125 following command could produce the output shown:

```
39126 val - <<EOF
39127 -y source s.x
39128 -m y s.y
39129 s.z
39130 EOF

39131 -y source s.x

39132 s.x: %Y%, -y mismatch
39133 s.z

39134 s.z: corrupted SCCS file
```

**39135 RATIONALE**

39136 None.

**39137 FUTURE DIRECTIONS**

39138 None.

**39139 SEE ALSO**

39140 *admin, delta, get, prs*

**39141 CHANGE HISTORY**

39142 First released in Issue 2.

**39143 Issue 4**

39144 Format reorganized.

39145 Exceptions to Utility Syntax Guidelines conformance noted.

39146 Internationalized environment variable support mandated.

**39147 Issue 6**

39148 The Open Group corrigenda item U025/4 has been applied, correcting a typographical error in  
39149 the EXIT STATUS.

## 39150 NAME

39151 vi — screen-oriented (visual) display editor

## 39152 SYNOPSIS

39153 UP `vi [-rR][-l][-c command][-t tagstring][-w size][file ...]`

39154

## 39155 DESCRIPTION

39156 **Notes to Reviewers**39157 *This section with side shading will not appear in the final copy. - Ed.*

39158 This utility has undergone significant revision due to the 1003.2b merger. The following D1  
 39159 XCU ERNs need to be checked after the merge: 361, 363, 364, 366.

39160 This utility shall be provided on systems that both support the User Portability Utilities option  
 39161 and define the POSIX2\_CHAR\_TERM symbol. On other systems it is optional.

39162 The *vi* (visual) utility is a screen-oriented text editor. Only the open and visual modes of the  
 39163 editor are described in IEEE Std. 1003.1-200x; see the line editor *ex* for additional editing  
 39164 capabilities used in *vi*. The user can switch back and forth between *vi* and *ex* and execute *ex*  
 39165 commands from within *vi*.

39166 This reference page uses the term *edit buffer* to describe the current working text. No specific  
 39167 implementation is implied by this term. All editing changes are performed on the edit buffer,  
 39168 and no changes to it shall affect any file until an editor command writes the file.

39169 When using *vi*, the terminal screen acts as a window into the editing buffer. Changes made to  
 39170 the editing buffer shall be reflected in the screen display; the position of the cursor on the screen  
 39171 shall indicate the position within the editing buffer.

39172 Certain terminals do not have all the capabilities necessary to support the complete *vi* definition.  
 39173 When these commands cannot be supported on such terminals, this condition shall not produce  
 39174 an error message such as “not an editor command” or report a syntax error. The implementation  
 39175 may either accept the commands and produce results on the screen that are the result of an  
 39176 unsuccessful attempt to meet the requirements of this volume of IEEE Std. 1003.1-200x or report  
 39177 an error describing the terminal-related deficiency.

## 39178 OPTIONS

39179 The *vi* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 39180 Section 12.2, Utility Syntax Guidelines.

39181 The following options shall be supported:

39182 **-c *command*** See the *ex* command description of the **-c** option.39183 MAN **-l** (The letter ell.) Set lisp mode; see **Edit Options in ex** on page 425.39184 **-r** See the *ex* command description of the **-r** option.39185 **-R** See the *ex* command description of the **-R** option.39186 **-t *tagstring*** See the *ex* command description of the **-t** option.39187 **-w *size*** See the *ex* command description of the **-w** option.

## 39188 OPERANDS

39189 See the OPERANDS section of the *ex* command for a description of the operands supported by  
 39190 the *vi* command.

**39191 STDIN**

39192 If standard input is not a terminal device, the results are undefined. The standard input consists  
 39193 of a series of commands and input text, as described in the EXTENDED DESCRIPTION section.

39194 If a read from the standard input returns an error, or if the editor detects an end-of-file condition  
 39195 from the standard input, it shall be equivalent to a SIGHUP asynchronous event.

**39196 INPUT FILES**

39197 See the INPUT FILES section of the *ex* command for a description of the input files supported by  
 39198 the *vi* command.

**39199 ENVIRONMENT VARIABLES**

39200 See the ENVIRONMENT VARIABLES section of the *ex* command for the environment variables  
 39201 that affect the execution of the *vi* command.

**39202 ASYNCHRONOUS EVENTS**

39203 See the ASYNCHRONOUS EVENTS section of the *ex* for the asynchronous events that affect the  
 39204 execution of the *vi* command.

**39205 STDOUT**

39206 If standard output is not a terminal device, undefined results occur.

39207 Standard output may be used for writing prompts to the user, for informational messages, and  
 39208 for writing lines from the file.

**39209 STDERR**

39210 If standard output is not a terminal device, undefined results occur.

39211 Used only for diagnostic messages.

**39212 OUTPUT FILES**

39213 See the OUTPUT FILES section of the *ex* command for a description of the output files  
 39214 supported by the *vi* command.

**39215 EXTENDED DESCRIPTION**

39216 If the terminal does not have the capabilities necessary to support an unspecified portion of the  
 39217 *vi* definition, implementations shall start initially in *ex* mode or open mode. Otherwise, after  
 39218 initialization, *vi* shall be in command mode; text input mode can be entered by one of several  
 39219 commands used to insert or change text. In text input mode, <ESC> can be used to return to  
 39220 command mode; other uses of <ESC> are described later in this section; see **Terminate**  
 39221 **Command or Input Mode** on page 1039.

**39222 Initialization in *ex* and *vi***

39223 See **Initialization in *ex* and *vi*** on page 392 for a description of *ex* and *vi* initialization for the *vi*  
 39224 utility.

**39225 Command Descriptions in *vi***

39226 The following symbols are used in this reference page to represent arguments to commands.

39227 *buffer* See the description of *buffer* in the EXTENDED DESCRIPTION section of the *ex* utility;  
 39228 see **Command Descriptions in *ex*** on page 401.

39229 In open and visual mode, when a command synopsis shows both [*buffer*] and [*count*]  
 39230 preceding the command name, they can be specified in either order.

39231 *count* A positive integer used as an optional argument to most commands, either to give a  
 39232 repeat count or as a size. This argument is optional and shall default to 1 unless  
 39233 otherwise specified.

The Synopsis lines for the *vi* commands `<control>-G`, `<control>-L`, `<control>-R`, `<control>-]`, `%`, `&`, `^`, `D`, `m`, `M`, `Q`, `u`, `U`, and `ZZ` do not have *count* as an optional argument. Regardless, it shall not be an error to specify a *count* to these commands, and any specified *count* shall be ignored.

***motion*** An optional trailing argument used by the `!`, `<`, `>`, `c`, `d`, and `y` commands, which is used to indicate the region of text that shall be affected by the command. The motion can be either one of the command characters repeated or one of several other *vi* commands (listed in the following table). Each of the applicable commands specifies the region of text matched by repeating the command; each command that can be used as a motion command specifies the region of text it affects.

Commands that take *motion* arguments operate on either lines or characters, depending on the circumstances. When operating on lines, all lines that fall partially or wholly within the text region specified for the command shall be affected. When operating on characters, only the exact characters in the specified text region shall be affected. Each motion command specifies this individually.

When commands that may be motion commands are not used as motion commands, they shall set the current position to the current line and column as specified.

The following commands shall be valid cursor motion commands:

|                                      |                 |                         |
|--------------------------------------|-----------------|-------------------------|
| <code>&lt;control&gt;-H</code>       | <code>;</code>  | <code>'character</code> |
| <code>&lt;newline&gt;</code>         | <code>?</code>  | <code>b</code>          |
| <code>&lt;carriage-return&gt;</code> | <code>B</code>  | <code>e</code>          |
| <code>&lt;control&gt;-N</code>       | <code>E</code>  | <code>f</code>          |
| <code>&lt;control&gt;-P</code>       | <code>F</code>  | <code>h</code>          |
| <code>&lt;space&gt;</code>           | <code>G</code>  | <code>j</code>          |
| <code>\$</code>                      | <code>H</code>  | <code>k</code>          |
| <code>%</code>                       | <code>L</code>  | <code>l</code>          |
| <code>'character</code>              | <code>M</code>  | <code>n</code>          |
| <code>(</code>                       | <code>N</code>  | <code>t</code>          |
| <code>)</code>                       | <code>T</code>  | <code>w</code>          |
| <code>+</code>                       | <code>W</code>  | <code>{</code>          |
| <code>,</code>                       | <code>[[</code> | <code> </code>          |
| <code>-</code>                       | <code>]]</code> | <code>}</code>          |
| <code>/</code>                       | <code>^</code>  | <code>0</code>          |
| <code>_</code>                       |                 |                         |

Any *count* that is specified to a command that has an associated motion command shall be applied to the motion command. If a *count* is applied to both the command and its associated motion command, the effect shall be multiplicative.

The following symbol is used in this section to specify locations in the edit buffer:

***current character***

The character that is currently displayed by the cursor.

The following symbols are used in this section to specify command actions:

***bigword*** In the POSIX locale, *vi* shall recognize four kinds of *bigwords*:

1. A maximal sequence of non-`<blank>` characters preceded and followed by `<blank>` characters or the beginning or end of a line or the edit buffer
2. One or more sequential empty or `<blank>` character-filled lines

|       |                                    |                                                                                                |  |
|-------|------------------------------------|------------------------------------------------------------------------------------------------|--|
| 39279 |                                    | 3. The first character in the edit buffer                                                      |  |
| 39280 |                                    | 4. The last character in the edit buffer                                                       |  |
| 39281 | <i>word</i>                        | In the POSIX locale, <i>vi</i> shall recognize five kinds of words:                            |  |
| 39282 |                                    | 1. A maximal sequence of letters, digits, and underscores, delimited at both ends by:          |  |
| 39283 |                                    | — Characters other than letters, digits, or underscores                                        |  |
| 39284 |                                    | — The beginning or end of a line                                                               |  |
| 39285 |                                    | — The beginning or end of the edit buffer                                                      |  |
| 39286 |                                    | 2. A maximal sequence of characters other than letters, digits, underscores, or                |  |
| 39287 |                                    | <blank> characters, delimited at both ends by:                                                 |  |
| 39288 |                                    | — A letter, digit, underscore                                                                  |  |
| 39289 |                                    | — <blank> characters                                                                           |  |
| 39290 |                                    | — The beginning or end of a line                                                               |  |
| 39291 |                                    | — The beginning or end of the edit buffer                                                      |  |
| 39292 |                                    | 3. One or more sequential empty or <blank> character-filled lines                              |  |
| 39293 |                                    | 4. The first character in the edit buffer                                                      |  |
| 39294 |                                    | 5. The last character in the edit buffer                                                       |  |
| 39295 | <i>section boundary</i>            |                                                                                                |  |
| 39296 |                                    | A <i>section boundary</i> is one of the following:                                             |  |
| 39297 |                                    | 1. A line whose first character is a <form-feed> character                                     |  |
| 39298 |                                    | 2. A line whose first character is an open curly brace ( ' { ' )                               |  |
| 39299 |                                    | 3. A line whose first character is a period and whose second and third characters              |  |
| 39300 |                                    | match a two-character pair in the <b>sections</b> edit option (see <i>ed</i> )                 |  |
| 39301 |                                    | 4. A line whose first character is a period and whose only other character matches             |  |
| 39302 |                                    | the first character of a two-character pair in the <b>sections</b> edit option, where the      |  |
| 39303 |                                    | second character of the two-character pair is a <space> character                              |  |
| 39304 |                                    | 5. The first line of the edit buffer                                                           |  |
| 39305 |                                    | 6. The last line of the edit buffer if the last line of the edit buffer is empty or if it is a |  |
| 39306 |                                    | ]] or } command; otherwise, the last character of the last line of the edit buffer             |  |
| 39307 | <i>paragraph boundary</i>          |                                                                                                |  |
| 39308 |                                    | A <i>paragraph boundary</i> is one of the following:                                           |  |
| 39309 |                                    | 1. A section boundary                                                                          |  |
| 39310 |                                    | 2. A line whose first character is a period and whose second and third characters              |  |
| 39311 |                                    | match a two-character pair in the <b>paragraphs</b> edit option (see <i>ed</i> )               |  |
| 39312 |                                    | 3. A line whose first character is a period and whose only other character matches             |  |
| 39313 |                                    | the first character of a two-character pair in the <i>paragraphs</i> edit option, where the    |  |
| 39314 |                                    | second character of the two-character pair is a <space> character                              |  |
| 39315 |                                    | 4. One or more sequential empty or <blank> character-filled lines                              |  |
| 39316 | <i>remembered search direction</i> |                                                                                                |  |
| 39317 |                                    | See the description of remembered search direction in <i>ed</i> .                              |  |

|       |                                                                                                               |  |
|-------|---------------------------------------------------------------------------------------------------------------|--|
| 39318 | <i>sentence boundary</i>                                                                                      |  |
| 39319 | A <i>sentence boundary</i> is one of the following:                                                           |  |
| 39320 | 1. A paragraph boundary                                                                                       |  |
| 39321 | 2. The first non-<blank> character that occurs after a paragraph boundary                                     |  |
| 39322 | 3. The first non-<blank> character that occurs after a period ( ' . ' ), exclamation                          |  |
| 39323 | point ( ' ! ' ), or question mark ( ' ? ' ), followed by two <space> characters or the                        |  |
| 39324 | end of a line; any number of closing parenthesis ( ' ) ' ), closing brackets ( ' ] ' ),                       |  |
| 39325 | double quote ( ' " ' ), ' or single quote ( ' ' ' ) characters can appear between the                         |  |
| 39326 | punctuation mark and the two <space> characters or end-of-line                                                |  |
| 39327 | Any lines displayed on the screen that logically represent lines after the last line in the edit buffer       |  |
| 39328 | shall consist of a single tilde ( ' ~ ' ) character.                                                          |  |
| 39329 | The last line of the screen shall be used to report errors or display informational messages. It              |  |
| 39330 | shall also be used to display the input for "line-oriented commands" ( / , ? , : , and ! ). When a line-      |  |
| 39331 | oriented command is executed, the editor shall enter text input mode on the last line on the                  |  |
| 39332 | screen, using the respective command characters as prompt characters. (In the case of the !                   |  |
| 39333 | command, the associated motion shall be entered by the user before the editor enters text input               |  |
| 39334 | mode.) The line entered by the user shall be terminated by a character, a non-<control>-V-                    |  |
| 39335 | escaped <carriage-return> character, or unescaped <ESC>. It is unspecified if more characters                 |  |
| 39336 | than require a display width minus one column number of screen columns can be entered.                        |  |
| 39337 | If any command is executed that overwrites a portion of the screen other than the last line of the            |  |
| 39338 | screen (for example, the <b>ex suspend</b> , or ! commands), other than the <b>ex shell</b> command, the user |  |
| 39339 | shall be prompted for a character before the screen is refreshed and the edit session continued.              |  |
| 39340 | <tab> characters shall take up the number of columns on the screen set by the <b>tabstop</b> edit             |  |
| 39341 | option (see <b>ed</b> ), unless there are less than that number of columns before the display margin that     |  |
| 39342 | will cause the displayed line to be folded; in this case, they shall only take up the number of               |  |
| 39343 | columns up to that boundary.                                                                                  |  |
| 39344 | The cursor shall be placed on the current line and relative to the current column as specified by             |  |
| 39345 | each command described in the following sections.                                                             |  |
| 39346 | In open mode, if the current line is not already displayed, then it shall be displayed.                       |  |
| 39347 | In visual mode, if the current line is not displayed, then the lines that are displayed shall be              |  |
| 39348 | expanded, scrolled, or redrawn to cause an unspecified portion of the current line to be                      |  |
| 39349 | displayed. If the screen is redrawn, no more than the number of logical lines specified by the                |  |
| 39350 | value of the <b>window</b> edit option shall be displayed (unless the current line cannot be completely       |  |
| 39351 | displayed in the number of logical lines specified by the <b>window</b> edit option) and the current          |  |
| 39352 | line shall be positioned as close to the center of the displayed lines as possible (within the                |  |
| 39353 | constraints imposed by the distance of the line from the beginning or end of the edit buffer). If             |  |
| 39354 | the current line is before the first line in the display and the screen is scrolled, an unspecified           |  |
| 39355 | portion of the current line shall be placed on the first line of the display. If the current line is after    |  |
| 39356 | the last line in the display and the screen is scrolled, an unspecified portion of the current line           |  |
| 39357 | shall be placed on the last line of the display.                                                              |  |
| 39358 | In visual mode, if a line from the edit buffer (other than the current line) does not entirely fit into       |  |
| 39359 | the lines at the bottom of the display that are available for its presentation, the editor may                |  |
| 39360 | choose not to display any portion of the line. The lines of the display that do not contain text              |  |
| 39361 | from the edit buffer for this reason shall each consist of a single ' @ ' character.                          |  |
| 39362 | In visual mode, the editor may choose for unspecified reasons to not update lines in the display              |  |
| 39363 | to correspond to the underlying edit buffer text. The lines of the display that do not correctly              |  |

39364 correspond to text from the edit buffer for this reason shall consist of a single '@' character, and  
 39365 the <control>-R command shall cause the editor to update the screen to correctly represent the  
 39366 edit buffer.

39367 Open and visual mode commands that set the current column set it to a column position in the  
 39368 display, and not a character position in the line. In this case, however, the column position in the  
 39369 display shall be calculated for a infinite width display; for example, the column related to a  
 39370 character that is part of a line that has been folded onto additional screen lines will be offset from  
 39371 the screen column where the physical line begins, not from the beginning of a particular screen  
 39372 line.

39373 The physical cursor column in the display is based on the value of the current column, as  
 39374 follows, with each rule applied in turn:

- 39375 1. If the current column is after the last screen column used by the displayed line, the  
 39376 physical cursor column shall be set to the last screen column occupied by the last character  
 39377 in the current line; otherwise, the physical cursor column shall be set to the current  
 39378 column.
- 39379 2. If the character of which some portion is displayed in the screen column specified by the  
 39380 physical cursor column requires more than a single screen column:
  - 39381 a. If in text input mode, the physical cursor column shall be adjusted to the first screen  
 39382 column in which any portion of that character is displayed.
  - 39383 b. Otherwise, the physical cursor column shall be adjusted to the last screen column in  
 39384 which any portion of that character is displayed.

39385 The current column shall not be changed by these adjustments to the physical cursor column.

39386 If an error occurs during the parsing or execution of a *vi* command:

- 39387 • The terminal shall be alerted. Execution of the *vi* command shall stop, and the cursor (for  
 39388 example, the current line and column) shall not be further modified.
- 39389 • Unless otherwise specified by the following command sections, it is unspecified whether an  
 39390 informational message shall be displayed.
- 39391 • Any partially entered *vi* command shall be discarded.
- 39392 • If the *vi* command resulted from a **map** expansion, all characters from that **map** expansion  
 39393 shall be discarded, except as otherwise specified by the **map** command (see *ed*).
- 39394 • If the *vi* command resulted from the execution of a buffer, no further commands caused by  
 39395 the execution of the buffer shall be executed.

## 39396 Page Backwards

39397 *Synopsis:*     [*count*] <control>-B

39398 If in open mode, the <control>-B command shall behave identically to the **z** command.  
 39399 Otherwise, if the current line is the first line of the edit buffer, it shall be an error.

39400 If the **window** edit option is less than 3, display a screen where the last line of the display shall  
 39401 be some portion of:

39402 (*current first line*) -1

39403 otherwise, display a screen where the first line of the display shall be some portion of:

39404 (*current first line*) -*count* W ((*window edit option*) -2)

39405 If this calculation would result in a line that is before the first line of the edit buffer, the first line  
 39406 of the display shall display some portion of the first line of the edit buffer.

39407 *Current line:* If no lines from the previous display remain on the screen, set to the last line of the  
 39408 display; otherwise, set to (*line* – the number of new lines displayed on this screen).

39409 *Current column:* Set to non-<blank>.

### 39410 **Scroll Forward**

39411 *Synopsis:* [ *count* ] <control>-D

39412 If the current line is the last line of the edit buffer, it shall be an error.

39413 If no *count* is specified, *count* shall default to the *count* associated with the previous <control>-D  
 39414 or <control>-U command. If there was no previous <control>-D or <control>-U command, *count*  
 39415 shall default to the value of the **scroll** edit option.

39416 If in open mode, write lines starting with the line after the current line, until *count* lines or the  
 39417 last line of the file have been written.

39418 *Current line:* If the current line + *count* is past the last line of the edit buffer, set to the last line of  
 39419 the edit buffer; otherwise, set to the current line + *count*.

39420 *Current column:* Set to non-<blank>.

### 39421 **Scroll Forward by Line**

39422 *Synopsis:* [ *count* ] <control>-E

39423 Display the line count lines after the last line currently displayed.

39424 If the last line of the edit buffer is displayed, it shall be an error. If there is no line *count* lines  
 39425 after the last line currently displayed, the last line of the display shall display some portion of  
 39426 the last line of the edit buffer.

39427 *Current line:* Unchanged if the previous current character is displayed; otherwise, set to the first  
 39428 line displayed.

39429 *Current column:* Unchanged.

### 39430 **Page Forward**

39431 *Synopsis:* [ *count* ] <control>-F

39432 If in open mode, the <control>-F command shall behave identically to the **z** command.  
 39433 Otherwise, if the current line is the last line of the edit buffer, it shall be an error.

39434 If the **window** edit option is less than 3, display a screen where the first line of the display shall  
 39435 be some portion of:

39436 (*current last line*) +1

39437 otherwise, display a screen where the first line of the display shall be some portion of:

39438 (*current first line*) + *count* W ((window edit option) -2)

39439 If this calculation would result in a line that is after the last line of the edit buffer, the last line of  
 39440 the display shall display some portion of the last line of the edit buffer.

39441 *Current line:* If no lines from the previous display remain on the screen, set to the first line of the  
 39442 display; otherwise, set to (*line* + the number of new lines displayed on this screen).



|       |                                                                                                                  |  |
|-------|------------------------------------------------------------------------------------------------------------------|--|
| 39443 | <i>Current column:</i> Set to non-<blank>.                                                                       |  |
| 39444 | <b>Display Information</b>                                                                                       |  |
| 39445 | <i>Synopsis:</i> <control>-G                                                                                     |  |
| 39446 | This command shall be equivalent to the <b>ex file</b> command .                                                 |  |
| 39447 | <b>Move Cursor Backwards</b>                                                                                     |  |
| 39448 | <i>Synopsis:</i> [ <i>count</i> ] <control>-H                                                                    |  |
| 39449 | [ <i>count</i> ] h                                                                                               |  |
| 39450 | the current erase character (see stty)                                                                           |  |
| 39451 | If there are no characters before the current character on the current line, it shall be an error. If            |  |
| 39452 | there are less than <i>count</i> previous characters on the current line, <i>count</i> shall be adjusted to the  |  |
| 39453 | number of previous characters on the line.                                                                       |  |
| 39454 | If used as a motion command:                                                                                     |  |
| 39455 | 1. The text region shall be from the character before the starting cursor up to and including                    |  |
| 39456 | the <i>count</i> th character before the starting cursor.                                                        |  |
| 39457 | 2. Any text copied to a buffer shall be in character mode.                                                       |  |
| 39458 | If not used as a motion command:                                                                                 |  |
| 39459 | <i>Current line:</i> Unchanged.                                                                                  |  |
| 39460 | <i>Current column:</i> Set to ( <i>column</i> – the number of columns occupied by <i>count</i> characters ending |  |
| 39461 | with the previous current column).                                                                               |  |
| 39462 | <b>Move Down</b>                                                                                                 |  |
| 39463 | <i>Synopsis:</i> [ <i>count</i> ] <newline>                                                                      |  |
| 39464 | [ <i>count</i> ] <control>-J                                                                                     |  |
| 39465 | [ <i>count</i> ] <control>-M                                                                                     |  |
| 39466 | [ <i>count</i> ] <control>-N                                                                                     |  |
| 39467 | [ <i>count</i> ] j                                                                                               |  |
| 39468 | [ <i>count</i> ] <carriage-return>                                                                               |  |
| 39469 | [ <i>count</i> ] +                                                                                               |  |
| 39470 | If there are less than <i>count</i> lines after the current line in the edit buffer, it shall be an error.       |  |
| 39471 | If used as a motion command:                                                                                     |  |
| 39472 | 1. The text region shall include the starting line and the next <i>count</i> – 1 lines.                          |  |
| 39473 | 2. Any text copied to a buffer shall be in line mode.                                                            |  |
| 39474 | If not used as a motion command:                                                                                 |  |
| 39475 | <i>Current line:</i> Set to <i>current line</i> + <i>count</i> .                                                 |  |
| 39476 | <i>Current column:</i> Set to non-<blank> for the <carriage-return> character, <control>-M, and +                |  |
| 39477 | commands; otherwise, unchanged.                                                                                  |  |

### 39478 **Clear and Redisplay**

39479 *Synopsis:*      <control>-L

39480 If in open mode, clear the screen and redisplay the current line. Otherwise, clear and redisplay  
39481 the screen.

39482 *Current line:* Unchanged.

39483 *Current column:* Unchanged.

### 39484 **Move Up**

39485 *Synopsis:*      [*count*] <control>-P

39486                    [*count*] k

39487                    [*count*] -

39488 If there are less than *count* lines before the current line in the edit buffer, it shall be an error.

39489 If used as a motion command:

39490      1. The text region shall include the starting line and the previous *count* lines.

39491      2. Any text copied to a buffer shall be in line mode.

39492 If not used as a motion command:

39493 *Current line:* Set to *current line* - *count*.

39494 *Current column:* Set to non-<blank> for the - command; otherwise, unchanged.

### 39495 **Redraw Screen**

39496 *Synopsis:*      <control>-R

39497 If any lines have been deleted from the logical screen and flagged as deleted on the terminal  
39498 using the @ convention (see the beginning of the EXTENDED DESCRIPTION section), they shall  
39499 be redisplayed to match the contents of the edit buffer.

39500 It is unspecified whether lines flagged with @ because they do not fit on the terminal display  
39501 shall be affected.

39502 *Current line:* Unchanged.

39503 *Current column:* Unchanged.

### 39504 **Scroll Backward**

39505 *Synopsis:*      [*count*] <control>-U

39506 If the current line is the first line of the edit buffer, it shall be an error.

39507 If no *count* is specified, *count* shall default to the *count* associated with the previous <control>-D  
39508 or <control>-U command. If there was no previous <control>-D or <control>-U command, *count*  
39509 shall default to the value of the **scroll** edit option.

39510 *Current line:* If *count* is greater than the current line, set to 1; otherwise, set to the current line -  
39511 *count*.

39512 *Current column:* Set to non-<blank>.

### 39513 **Scroll Backward by Line**

39514 *Synopsis:*     [*count*] <control>-Y

39515 Display the line *count* lines before the first line currently displayed.

39516 If the current line is the first line of the edit buffer, it shall be an error. If this calculation would  
39517 result in a line that is before the first line of the edit buffer, the first line of the display shall  
39518 display some portion of the first line of the edit buffer.

39519 *Current line:* Unchanged if the previous current character is displayed; otherwise, set to the first  
39520 line displayed.

39521 *Current column:* Unchanged.

### 39522 **Edit the Alternate File**

39523 *Synopsis:*     <control>-^

39524 This command shall be equivalent to the *ex* **edit** command, with the alternate path name as its  
39525 argument.

### 39526 **Terminate Command or Input Mode**

39527 *Synopsis:*     <ESC>

39528 If a partial *vi* command (as defined by at least one, non-*count* character) has been entered,  
39529 discard the *count* and the command character(s).

39530 Otherwise, if no command characters have been entered, and the <ESC> was the result of a map  
39531 expansion, the terminal shall be alerted and the <ESC> character shall be discarded, but it shall  
39532 not be an error.

39533 Otherwise, it shall be an error.

39534 *Current line:* Unchanged.

39535 *Current column:* Unchanged.

### 39536 **Search for tagstring**

39537 *Synopsis:*     <control>-]

39538 If the current character is not a word or <blank> character, it shall be an error.

39539 This command shall be equivalent to the *ex* **tag** command, with the argument to that command  
39540 defined as follows.

39541 If the current character is a <blank> character:

- 39542     1. Skip all <blank> characters after the cursor up to the end of the line.
- 39543     2. If the end of the line is reached, it shall be an error.

39544 Then, the argument to the *ex* **tag** command shall be the current character and all subsequent  
39545 characters, up to the first non-word character or the end of the line.

**Move Cursor Forward**

*Synopsis:*     [ *count* ] <space>  
                   [ *count* ] 1 (ell)

If there are less than *count* characters after the cursor on the current line, *count* shall be adjusted to the number of characters after the cursor on the line.

If used as a motion command:

1. If the current or *count*th character after the cursor is the last character in the line, the text region shall be comprised of the current character up to and including the last character in the line. Otherwise, the text region shall be from the current character up to, but not including, the *count*th character after the cursor.
2. Any text copied to a buffer shall be in character mode.

If not used as a motion command:

If there are no characters after the current character on the current line, it shall be an error.

*Current line:* Unchanged.

*Current column:* Set to the last column that displays any portion of the *count*th character after the current character.

**Replace Text with Results from Shell Command**

*Synopsis:*     [ *count* ] ! *motion shell-commands* <newline>

If the motion command is the ! command repeated:

1. If the edit buffer is empty and no *count* was supplied, the command shall be the equivalent of the *ex:read!* command, with the text input, and no text shall be copied to any buffer.
2. Otherwise:
  - a. If there are less than *count* – 1 lines after the current line in the edit buffer, it shall be an error.
  - b. The text region shall be from the current line up to and including the next *count* – 1 lines.

Otherwise, the text region shall be the lines in which any character of the text region specified by the motion command appear.

Any text copied to a buffer shall be in line mode.

This command shall be equivalent to the *ex!* command for the specified lines.

**Move Cursor to End-of-line**

*Synopsis:*     [ *count* ] \$

It shall be an error if there are less than (*count* – 1) lines after the current line in the edit buffer.

If used as a motion command:

1. If *count* is 1:
  - a. It shall be an error if the line is empty.
  - b. Otherwise, the text region shall consist of all characters from the starting cursor to the last character in the line, inclusive, and any text copied to a buffer shall be in

|       |                                                                                                          |  |
|-------|----------------------------------------------------------------------------------------------------------|--|
| 39584 | character mode.                                                                                          |  |
| 39585 | 2. Otherwise, if the starting cursor position is at or before the first non-<blank> character in         |  |
| 39586 | the line, the text region shall consist of the current and the next <i>count</i> -1 lines, and any text  |  |
| 39587 | saved to a buffer shall be in line mode.                                                                 |  |
| 39588 | 3. Otherwise, the text region shall consist of all characters from the starting cursor to the last       |  |
| 39589 | character in the line that is <i>count</i> -1 lines forward from the current line, and any text copied   |  |
| 39590 | to a buffer shall be in character mode.                                                                  |  |
| 39591 | If not used as a motion command:                                                                         |  |
| 39592 | <i>Current line</i> : Set to the <i>current line</i> + <i>count</i> -1.                                  |  |
| 39593 | <i>Current column</i> : The current column is set to the last screen column of the last character in the |  |
| 39594 | line, or column position 1 if the line is empty.                                                         |  |
| 39595 | The current column shall be adjusted to be on the last screen column of the last character of the        |  |
| 39596 | current line as subsequent commands change the current line, until a command changes the                 |  |
| 39597 | current column.                                                                                          |  |
| 39598 | <b>Move to Matching Character</b>                                                                        |  |
| 39599 | <i>Synopsis</i> :      %                                                                                 |  |
| 39600 | If the character at the current position is not a parenthesis, bracket, or curly brace, search           |  |
| 39601 | forwards in the line to the first one of those characters. If no such character is found, it shall be    |  |
| 39602 | an error.                                                                                                |  |
| 39603 | The matching character shall be the parenthesis, bracket, or curly brace matching the                    |  |
| 39604 | parenthesis, bracket, or curly brace, respectively, that was at the current position or that was         |  |
| 39605 | found on the current line.                                                                               |  |
| 39606 | Matching shall be determined as follows, for an open parenthesis:                                        |  |
| 39607 | 1. Set a counter to 1.                                                                                   |  |
| 39608 | 2. Search forwards until a parenthesis is found or the end of the edit buffer is reached.                |  |
| 39609 | 3. If the end of the edit buffer is reached, it shall be an error.                                       |  |
| 39610 | 4. If an open parenthesis is found, increment the counter by 1.                                          |  |
| 39611 | 5. If a close parenthesis is found, decrement the counter by 1.                                          |  |
| 39612 | 6. If the counter is zero, the current character is the matching character.                              |  |
| 39613 | Matching for a close parenthesis shall be equivalent, except that the search shall be backwards,         |  |
| 39614 | from the starting character to the beginning of the buffer, a close parenthesis shall increment the      |  |
| 39615 | counter by 1, and an open parenthesis shall decrement the counter by 1.                                  |  |
| 39616 | Matching for brackets and curly braces shall be equivalent, except that searching shall be done          |  |
| 39617 | for open and close brackets or open and close curly braces. It is implementation-dependent               |  |
| 39618 | whether other characters are searched for and matched as well.                                           |  |
| 39619 | If used as a motion command:                                                                             |  |
| 39620 | 1. If the matching cursor was after the starting cursor in the edit buffer, and the starting             |  |
| 39621 | cursor position was at or before the first non-<blank> character in the starting line, and the           |  |
| 39622 | matching cursor position was at or after the last non-<blank> character in the matching                  |  |
| 39623 | line, the text region shall consist of the current line to the matching line, inclusive, and any         |  |
| 39624 | text copied to a buffer shall be in line mode.                                                           |  |

39625 2. If the matching cursor was before the starting cursor in the edit buffer, and the starting  
 39626 cursor position was at or after the last non-<blank> character in the starting line, and the  
 39627 matching cursor position was at or before the first non-<blank> character in the matching  
 39628 line, the text region shall consist of the current line to the matching line, inclusive, and any  
 39629 text copied to a buffer shall be in line mode.

39630 3. Otherwise, the text region shall consist of the starting character to the matching character,  
 39631 inclusive, and any text copied to a buffer shall be in character mode.

39632 If not used as a motion command:

39633 *Current line*: Set to the line where the matching character is located.

39634 *Current column*: Set to the last column where any portion of the matching character is displayed.

### 39635 **Repeat Substitution**

39636 *Synopsis*:        &

39637 Repeat the previous substitution command. This command shall be equivalent to the *ex &*  
 39638 command with the current line as its addresses, and without *options*, *count*, or *flags*.

### 39639 **Return to Previous Context at Beginning of Line**

39640 *Synopsis*:        ' *character*

39641 It shall be an error if there is no line in the edit buffer marked by *character*.

39642 If used as a motion command:

39643 1. If the starting cursor is after the marked cursor, then the locations of the starting cursor  
 39644 and the marked cursor in the edit buffer shall be logically swapped.

39645 2. The text region shall consist of the starting line up to and including the marked line, and  
 39646 any text copied to a buffer shall be in line mode.

39647 If not used as a motion command:

39648 *Current line*: Set to the line referenced by the mark.

39649 *Current column*: Set to non-<blank>.

### 39650 **Return to Previous Context**

39651 *Synopsis*:        ` *character*

39652 It shall be an error if the marked line is no longer in the edit buffer. If the marked line no longer  
 39653 contains a character in the saved numbered character position, it shall be as if the marked  
 39654 position is the first non-<blank> character.

39655 If used as a motion command:

39656 1. It shall be an error if the marked cursor references the same character in the edit buffer as  
 39657 the starting cursor.

39658 2. If the starting cursor is after the marked cursor, then the locations of the starting cursor  
 39659 and the marked cursor in the edit buffer shall be logically swapped.

39660 3. If the starting line is empty or the starting cursor is at or before the first non-<blank>  
 39661 character of the starting line, and the marked cursor line is empty or the marked cursor  
 39662 references the first character of the marked cursor line, the text region shall consist of all  
 39663 lines containing characters from the starting cursor to the line before the marked cursor

|       |                                                                                                                  |  |
|-------|------------------------------------------------------------------------------------------------------------------|--|
| 39664 | line, inclusive, and any text copied to a buffer shall be in line mode.                                          |  |
| 39665 | 4. Otherwise, if the marked cursor line is empty or the marked cursor references a character                     |  |
| 39666 | at or before the first non-<blank> character of the marked cursor line, the region of text                       |  |
| 39667 | shall be from the starting cursor to the last character of the line before the marked cursor                     |  |
| 39668 | line, inclusive, and any text copied to a buffer shall be in character mode.                                     |  |
| 39669 | 5. Otherwise, the region of text shall be from the starting cursor (inclusive), to the marked                    |  |
| 39670 | cursor (exclusive), and any text copied to a buffer shall be in character mode.                                  |  |
| 39671 | If not used as a motion command:                                                                                 |  |
| 39672 | <i>Current line</i> : Set to the line referenced by the mark.                                                    |  |
| 39673 | <i>Current column</i> : Set to the last column in which any portion of the character referenced by the           |  |
| 39674 | mark is displayed.                                                                                               |  |
| 39675 | <b>Return to Previous Section</b>                                                                                |  |
| 39676 | <i>Synopsis:</i> [ [                                                                                             |  |
| 39677 | Move the cursor backward through the edit buffer to the first character of the previous section                  |  |
| 39678 | boundary, <i>count</i> times.                                                                                    |  |
| 39679 | If used as a motion command:                                                                                     |  |
| 39680 | 1. If the starting cursor was at the first character of the starting line or the starting line was               |  |
| 39681 | empty, and the first character of the boundary was the first character of the boundary line,                     |  |
| 39682 | the text region shall consist of the current line up to and including the line where the                         |  |
| 39683 | <i>countth</i> next boundary starts, and any text copied to a buffer shall be in line mode.                      |  |
| 39684 | 2. If the boundary was the last line of the edit buffer or the last character of the last line of the            |  |
| 39685 | edit buffer, the text region shall consist of the last character in the edit buffer up to and                    |  |
| 39686 | including the starting character, and any text saved to a buffer shall be in character mode.                     |  |
| 39687 | 3. Otherwise, the text region shall consist of the starting character up to but not including the                |  |
| 39688 | first character in the <i>countth</i> next boundary, and any text copied to a buffer shall be in                 |  |
| 39689 | character mode.                                                                                                  |  |
| 39690 | MAN If the <i>lisp</i> option is set, a section boundary is also identified by a line with a leading ' ( '.      |  |
| 39691 | If not used as a motion command:                                                                                 |  |
| 39692 | <i>Current line</i> : Set to the line where the <i>countth</i> next boundary in the edit buffer starts.          |  |
| 39693 | <i>Current column</i> : Set to the last column in which any portion of the first character of the <i>countth</i> |  |
| 39694 | next boundary is displayed, or column position 1 if the line is empty.                                           |  |
| 39695 | <b>Move to Next Section</b>                                                                                      |  |
| 39696 | <i>Synopsis:</i> ] ]                                                                                             |  |
| 39697 | Move the cursor forward through the edit buffer to the first character of the next section                       |  |
| 39698 | boundary, <i>count</i> times.                                                                                    |  |
| 39699 | If used as a motion command:                                                                                     |  |
| 39700 | 1. If the starting cursor was at the first character of the starting line or the starting line was               |  |
| 39701 | empty, and the first character of the boundary was the first character of the boundary line,                     |  |
| 39702 | the text region shall consist of the current line up to and including the line where the                         |  |
| 39703 | <i>countth</i> previous boundary starts, and any text copied to a buffer shall be in line mode.                  |  |

|       |     |                                                                                                                   |
|-------|-----|-------------------------------------------------------------------------------------------------------------------|
| 39704 | 2.  | If the boundary was the first line of the edit buffer, the text region shall consist of the first                 |
| 39705 |     | character in the edit buffer up to but not including the starting character, and any text                         |
| 39706 |     | copied to a buffer shall be in character mode.                                                                    |
| 39707 | 3.  | Otherwise, the text region shall consist of the first character in the <i>count</i> th previous section           |
| 39708 |     | boundary up to but not including the starting character, and any text copied to a buffer                          |
| 39709 |     | shall be in character mode.                                                                                       |
| 39710 | MAN | If the <i>lisp</i> option is set, a section boundary is also identified by a line with a leading ' ( ' .          |
| 39711 |     | If not used as a motion command:                                                                                  |
| 39712 |     | <i>Current line</i> : Set to the line where the <i>count</i> th previous boundary in the edit buffer starts.      |
| 39713 |     | <i>Current column</i> : Set to the last column in which any portion of the first character of the <i>count</i> th |
| 39714 |     | previous boundary is displayed, or column position 1 if the line is empty.                                        |
| 39715 |     | <b>Move to First Non-&lt;blank&gt; Position on Current Line</b>                                                   |
| 39716 |     | <i>Synopsis</i> :     ^                                                                                           |
| 39717 |     | If used as a motion command:                                                                                      |
| 39718 | 1.  | If the line has no non-<blank> characters, or if the cursor is at the first non-<blank>                           |
| 39719 |     | character of the line, it shall be an error.                                                                      |
| 39720 | 2.  | If the cursor is before the first non-<blank> character of the line, the text region shall be                     |
| 39721 |     | comprised of the current character, up to, but not including, the first non-<blank>                               |
| 39722 |     | character of the line.                                                                                            |
| 39723 | 3.  | If the cursor is after the first non-<blank> character of the line, the text region shall be from                 |
| 39724 |     | the character before the starting cursor up to and including the first non-<blank> character                      |
| 39725 |     | of the line.                                                                                                      |
| 39726 | 4.  | Any text copied to a buffer shall be in character mode.                                                           |
| 39727 |     | If not used as a motion command:                                                                                  |
| 39728 |     | <i>Current line</i> : Unchanged.                                                                                  |
| 39729 |     | <i>Current column</i> : Set to non-<blank>.                                                                       |
| 39730 |     | <b>Current and line above</b>                                                                                     |
| 39731 |     | <i>Synopsis</i> :     [ <i>count</i> ] _                                                                          |
| 39732 |     | If there are less than <i>count</i> – 1 lines after the current line in the edit buffer, it shall be an error.    |
| 39733 |     | If used as a motion command:                                                                                      |
| 39734 | 1.  | If <i>count</i> is less than 2, the text region shall be the current line.                                        |
| 39735 | 2.  | Otherwise, the text region shall include the starting line and the next <i>count</i> – 1 lines.                   |
| 39736 | 3.  | Any text copied to a buffer shall be in line mode.                                                                |
| 39737 |     | If not used as a motion command:                                                                                  |
| 39738 |     | <i>Current line</i> : Set to current line + <i>count</i> – 1.                                                     |
| 39739 |     | <i>Current column</i> : Set to non-<blank>.                                                                       |



### 39740 **Move Back to Beginning of Sentence**

39741 *Synopsis:*     [ *count* ] (

39742     Move backward to the beginning of a sentence. This command shall be equivalent to the `[]`  
39743     command, with the exception that sentence boundaries shall be used instead of section  
39744     boundaries.

39745 MAN     If the *lisp* option is set, a *lisp* s-expression is considered a sentence for this command.

### 39746 **Move Forward to Beginning of Sentence**

39747 *Synopsis:*     [ *count* ] )

39748     Move forward to the beginning of a sentence. This command shall be equivalent to the `]]`  
39749     command, with the exception that sentence boundaries shall be used instead of section  
39750     boundaries.

39751 MAN     If the *lisp* option is set, a *lisp* s-expression is considered a sentence for this command.

### 39752 **Move Back to Preceding Paragraph**

39753 *Synopsis:*     [ *count* ] {

39754     Move back to the beginning of the preceding paragraph. This command shall be equivalent to  
39755     the `[]` command, with the exception that paragraph boundaries shall be used instead of section  
39756     boundaries.

### 39757 **Move Forward to Next Paragraph**

39758 *Synopsis:*     [ *count* ] }

39759     Move forward to the beginning of the next paragraph. This command shall be equivalent to the  
39760     `]]` command, with the exception that paragraph boundaries shall be used instead of section  
39761     boundaries.

### 39762 **Move to Specific Column Position**

39763 *Synopsis:*     [ *count* ] |

39764     For the purposes of this command, lines that are too long for the current display and that have  
39765     been folded shall be treated as having a single, 1-based, number of columns.

39766     If there are less than *count* columns in which characters from the current line are displayed on  
39767     the screen, *count* shall be adjusted to be the last column in which any portion of the line is  
39768     displayed on the screen.

39769     If used as a motion character:

- 39770         1. If the line is empty, or the cursor character is the same as the character on the *count*th  
39771         column of the line, it shall be an error.
- 39772         2. If the cursor is before the *count*th column of the line, the text region shall be comprised of  
39773         the current character, up to but not including the character on the *count*th column of the  
39774         line.
- 39775         3. If the cursor is after the *count*th column of the line, the text region shall be from the  
39776         character before the starting cursor up to and including the character on the *count*th  
39777         column of the line.

39778 4. Any text copied to a buffer shall be in character mode.  
 39779 If not used as a motion character:  
 39780 *Current line*: Unchanged.  
 39781 *Current column*: Set to the last column in which any portion of the character that is displayed in  
 39782 the *count* column of the line is displayed.

### 39783 **Reverse Find Character**

39784 *Synopsis*: [ *count* ] ,  
 39785 If the last **F**, **f**, **T**, or **t** command was **F**, **f**, **T**, or **t**, this command shall be equivalent to an **f**, **F**, **t**, or  
 39786 **T** command, respectively, with the specified *count* and the same search character.  
 39787 If there was no previous **F**, **f**, **T**, or **t** command, it shall be an error.

### 39788 **Repeat**

39789 *Synopsis*: [ *count* ] .  
 39790 Repeat the last **!**, **<**, **>**, **A**, **C**, **D**, **I**, **J**, **O**, **P**, **R**, **S**, **X**, **Y**, **a**, **c**, **d**, **i**, **o**, **p**, **r**, **s**, **x**, **y**, or **~** command. It shall  
 39791 be an error if none of these commands have been executed. Commands (other than commands  
 39792 that enter text input mode) executed as a result of map expansions, shall not change the value of  
 39793 the last repeatable command.  
 39794 Repeated commands with associated motion commands shall repeat the motion command as  
 39795 well; however, any specified *count* shall replace the *count*(s) that were originally specified to the  
 39796 repeated command or its associated motion command.  
 39797 If the motion component of the repeated command is **f**, **F**, **t**, or **T**, the repeated command shall  
 39798 not set the remembered search character for the **;** and **,** commands.  
 39799 If the repeated command is **p** or **P**, and the buffer associated with that command was a numeric  
 39800 buffer named with a number less than 9, the buffer associated with the repeated command shall  
 39801 be set to be the buffer named by the name of the previous buffer logically incremented by 1.  
 39802 If the repeated character is a text input command, the input text associated with that command  
 39803 is repeated literally:  
 39804 • Input characters are neither macro or abbreviation-expanded.  
 39805 • Input characters are not interpreted in any special way with the exception that the <newline>  
 39806 character and the <carriage-return> character, and <control>-T behave as described in **Input**  
 39807 **Mode Commands in vi** on page 1064.  
 39808 *Current line*: Set as described for the repeated command.  
 39809 *Current column*: Set as described for the repeated command.

### 39810 **Find Regular Expression**

39811 *Synopsis*: /  
 39812 If the input line contains no characters, it shall be equivalent to a line containing only the last  
 39813 regular expression encountered. The enhanced regular expressions supported by **vi** are  
 39814 described in **Regular Expressions in ex** on page 424.  
 39815 Otherwise, the line shall be interpreted as one or more regular expressions, optionally followed  
 39816 by an address offset or a **vi z** command.

39817 If the regular expression is not the last regular expression on the line, or if a line offset or **z**  
 39818 command is specified, the regular expression shall be terminated by an unescaped `'/'`  
 39819 character, which shall not be used as part of the regular expression. If the regular expression is  
 39820 not the first regular expression on the line, it shall be preceded by zero or more `<blank>`  
 39821 characters, a semicolon, zero or more `<blank>` characters, and a leading `'/'` character, which  
 39822 shall not be interpreted as part of the regular expression. It shall be an error to precede any  
 39823 regular expression with any characters other than these.

39824 Each search shall begin from the character after the first character of the last match (or, if it is the  
 39825 first search, after the cursor). If the **wraps** edit option is set, the search shall continue to the  
 39826 character before the starting cursor character; otherwise, to the end of the edit buffer. It shall be  
 39827 an error if any search fails to find a match, and an informational message to this effect shall be  
 39828 displayed.

39829 An optional address offset (see **Addressing in ex** on page 394) can be specified after the last  
 39830 regular expression by including a trailing `'/'` character after the regular expression and  
 39831 specifying the address offset. This offset will be from the line containing the match for the last  
 39832 regular expression specified. It shall be an error if the line offset would indicate a line address  
 39833 less than 1 or greater than the last line in the edit buffer. An address offset of zero shall be  
 39834 supported. It shall be an error to follow the address offset with any other characters than  
 39835 `<blank>` characters.

39836 If not used as a motion command, an optional **z** command (see **Redraw Window** on page 1063)  
 39837 can be specified after the last regular expression by including a trailing `'/'` character after the  
 39838 regular expression, zero or more `<blank>` characters, a `'z'`, zero or more `<blank>` characters, an  
 39839 optional new **window** edit option value, zero or more `<blank>` characters, and a location  
 39840 character. The effect shall be as if the **z** command was executed after the `/` command. It shall be  
 39841 an error to follow the **z** command with any other characters than `<blank>` characters.

39842 The remembered search direction shall be set to forward.

39843 If used as a motion command:

- 39844 1. It shall be an error if the last match references the same character in the edit buffer as the  
 39845 starting cursor.
- 39846 2. If any address offset is specified, the last match shall be adjusted by the specified offset as  
 39847 described previously.
- 39848 3. If the starting cursor is after the last match, then the locations of the starting cursor and the  
 39849 last match in the edit buffer shall be logically swapped.
- 39850 4. If any address offset is specified, the text region shall consist of all lines containing  
 39851 characters from the starting cursor to the last match line, inclusive, and any text copied to a  
 39852 buffer shall be in line mode.
- 39853 5. Otherwise, if the starting line is empty or the starting cursor is at or before the first non-  
 39854 `<blank>` character of the starting line, and the last match line is empty or the last match  
 39855 starts at the first character of the last match line, the text region shall consist of all lines  
 39856 containing characters from the starting cursor to the line before the last match line,  
 39857 inclusive, and any text copied to a buffer shall be in line mode.
- 39858 6. Otherwise, if the last match line is empty or the last match begins at a character at or  
 39859 before the first non-`<blank>` of the last match line, the region of text shall be from the  
 39860 current cursor to the last character of the line before the last match line, inclusive, and any  
 39861 text copied to a buffer shall be in character mode.

39862 7. Otherwise, the region of text shall be from the current cursor (inclusive), to the first  
 39863 character of the last match (exclusive), and any text copied to a buffer shall be in  
 39864 character mode.

39865 If not used as a motion command:

39866 *Current line*: If a match is found, set to the last matched line plus the address offset, if any;  
 39867 otherwise, unchanged.

39868 *Current column*: Set to the last column on which any portion of the first character in the last  
 39869 matched string is displayed, if a match is found; otherwise, unchanged.

### 39870 **Move to First Character in Line**

39871 *Synopsis*: 0 (zero)

39872 Move to the first character on the current line. The character '0' shall not be interpreted as a  
 39873 command if it is immediately preceded by a digit.

39874 If used as a motion command:

- 39875 1. If the cursor character is the first character in the line, it shall be an error.
- 39876 2. The text region shall be from the character before the cursor character up to and including  
 39877 the first character in the line.
- 39878 3. Any text copied to a buffer shall be in character mode.

39879 If not used as a motion command:

39880 *Current line*: Unchanged.

39881 *Current column*: The last column in which any portion of the first character in the line is  
 39882 displayed, or if the line is empty, unchanged.

### 39883 **Execute an ex Command**

39884 *Synopsis*: :

39885 Execute one or more *ex* commands.

39886 If any portion of the screen other than the last line of the screen was overwritten by any *ex*  
 39887 command (except **shell**), *vi* shall display a message indicating that it is waiting for an input from  
 39888 the user, and shall then read a character. This action may also be taken for other, unspecified  
 39889 reasons.

39890 If the next character entered is a ' : ', another *ex* command shall be accepted and executed. Any  
 39891 other character shall cause the screen to be refreshed and *vi* shall return to command mode.

39892 *Current line*: As specified for the *ex* command.

39893 *Current column*: As specified for the *ex* command.

39894 **Repeat Find**39895 *Synopsis:* [count] ;

39896 This command shall be equivalent to the last **F**, **f**, **T**, or **t** command, with the specified *count*, and  
 39897 with the same search character used for the last **F**, **f**, **T**, or **t** command. If there was no previous **F**,  
 39898 **f**, **T**, or **t** command, it shall be an error.

39899 **Shift Left**39900 *Synopsis:* [count] < motion

39901 If the motion command is the &lt; command repeated:

39902 1. If there are less than *count* -1 lines after the current line in the edit buffer, it shall be an  
 39903 error.

39904 2. The text region shall be from the current line, up to and including the next *count* -1 lines.

39905 Shift any line in the text region specified by the *count* and motion command one shiftwidth (see  
 39906 the *ex* **shiftwidth** option) toward the start of the line, as described by the *ex* < command. The  
 39907 unshifted lines shall be copied to the unnamed buffer in line mode.

39908 *Current line:* If the motion was from the current cursor position toward the end of the edit  
 39909 buffer, unchanged. Otherwise, set to the first line in the edit buffer that is part of the text region  
 39910 specified by the motion command.

39911 *Current column:* Set to non-<blank>.39912 **Shift Right**39913 *Synopsis:* [count] > motion

39914 If the motion command is the &gt; command repeated:

39915 1. If there are less than *count* -1 lines after the current line in the edit buffer, it shall be an  
 39916 error.

39917 2. The text region shall be from the current line, up to and including the next *count* -1 lines.

39918 Shift any line with characters in the text region specified by the *count* and motion command one  
 39919 shiftwidth (see the *ex* **shiftwidth** option) away from the start of the line, as described by the *ex* >  
 39920 command. The unshifted lines shall be copied into the unnamed buffer in line mode.

39921 *Current line:* If the motion was from the current cursor position toward the end of the edit  
 39922 buffer, unchanged. Otherwise, set to the first line in the edit buffer that is part of the text region  
 39923 specified by the motion command.

39924 *Current column:* Set to non-<blank>.39925 **Scan Backwards for Regular Expression**39926 *Synopsis:* ?

39927 Scan backwards; The ? command shall be equivalent to the / command (see **Find Regular**  
 39928 **Expression** on page 1046) with the following exceptions:

39929 1. The input prompt shall be a ' ? '.

39930 2. Each search shall begin from the character before the first character of the last match (or, if  
 39931 it is the first search, the character before the cursor character).

- 39932 3. The search direction shall be from the cursor toward the beginning of the edit buffer, and  
 39933 the **wraps**can edit option shall affect whether the search wraps to the end of the edit buffer  
 39934 and continues.
- 39935 4. The remembered search direction shall be set to backward.

### 39936 **Execute**

39937 *Synopsis:*     @*buffer*

39938 If the *buffer* is specified as @, the last buffer executed shall be used. If no previous buffer has been  
 39939 executed, it shall be an error.

39940 Behave as if the contents of the named buffer were entered as standard input. After each line of a  
 39941 line-mode buffer, and all but the last line of a character mode buffer, behave as if a <newline>  
 39942 character were entered as standard input.

39943 If an error occurs during this process, an error message shall be written, and no more characters  
 39944 resulting from the execution of this command shall be processed.

39945 If a *count* is specified, behave as if that count were entered as user input before the characters  
 39946 from the @ buffer were entered.

39947 *Current line:* As specified for the individual commands.

39948 *Current column:* As specified for the individual commands.

### 39949 **Reverse Case**

39950 *Synopsis:*     [*count*] ~

39951 Reverse the case of the current character and the next *count* –1 characters, such that lowercase  
 39952 characters that have uppercase counterparts shall be changed to uppercase characters, and  
 39953 uppercase characters that have lowercase counterparts shall be changed to lowercase characters,  
 39954 as prescribed by the current locale. No other characters shall be affected by this command.

39955 If there are less than *count* –1 characters after the cursor in the edit buffer, *count* shall be adjusted  
 39956 to the number of characters after the cursor in the edit buffer minus 1.

39957 For the purposes of this command, the next character after the last character on the line shall be  
 39958 the next character in the edit buffer.

39959 *Current line:* Set to the line including the (*count*–1)th character after the cursor.

39960 *Current column:* Set to the last column in which any portion of the (*count*–1)th character after the  
 39961 cursor is displayed.

### 39962 **Reindent**

39963 MAN *Synopsis:*

39964                   [*count*]=[*motion*]

39965 If the *lisp* option is set, reindents the specified lines, as though they were typed in with **lisp** and  
 39966 **autoindent** set.

39967 *Current line:* Unchanged.

39968 *Current column:* Move to the first non-<blank> character of the line or the last character if the  
 39969 line is a blank line.

39970      **Append**39971      *Synopsis:*      [ *count* ] a

39972      Enter text input mode after the current cursor position. No characters already in the edit buffer  
 39973      shall be affected by this command. A *count* shall cause the input text to be appended *count* -1  
 39974      more times to the end of the input.

39975      *Current line/column:* As specified for the text input commands (see **Input Mode Commands in vi**  
 39976      on page 1064).

39977      **Append at End-of-Line**39978      *Synopsis:*      [ *count* ] A39979      This command shall be equivalent to the *vi* command:39980      \$ [ *count* ] a39981      (see **Append**).39982      **Move Backward to Preceding Word**39983      *Synopsis:*      [ *count* ] b

39984      With the exception that words are used as the delimiter instead of bigwords, this command shall  
 39985      be equivalent to the **B** command.

39986      **Move Backward to Preceding Bigword**39987      *Synopsis:*      [ *count* ] B

39988      If the edit buffer is empty or the cursor is on the first character of the edit buffer, it shall be an  
 39989      error. If less than *count* bigwords begin between the cursor and the start of the edit buffer, *count*  
 39990      shall be adjusted to the number of bigword beginnings between the cursor and the start of the  
 39991      edit buffer.

39992      If used as a motion command:

39993      1. The text region shall be from the first character of the *count*th previous bigword beginning  
 39994      up to but not including the cursor character.

39995      2. Any text copied to a buffer shall be in character mode.

39996      If not used as a motion command:

39997      *Current line:* Set to the line containing the *current column*.

39998      *Current column:* Set to the last column upon which any part of the first character of the *count*th  
 39999      previous bigword is displayed.

40000      **Change**40001      *Synopsis:*      [ *buffer* ] [ *count* ] c *motion*40002      If the motion command is the *c* command repeated:

40003      1. The buffer text shall be in line mode.

40004      2. If there are less than *count* -1 lines after the current line in the edit buffer, it shall be an  
 40005      error.

40006           3. The text region shall be from the current line up to and including the next *count* –1 lines. |  
40007 Otherwise, the buffer text mode and text region shall be as specified by the motion command. |  
40008 The replaced text shall be copied into *buffer*, if specified, and into the unnamed buffer. If the text |  
40009 to be replaced contains characters from more than a single line, or the buffer text is in line mode, |  
40010 the replaced text shall be copied into the numeric buffers as well. |  
40011 If the buffer text is in line mode: |  
40012           1. Any lines that contain characters in the region shall be deleted, and the editor shall enter |  
40013 text input mode at the beginning of a new line which shall replace the first line deleted. |  
40014           2. If the **autoindent** edit option is set, **autoindent** characters equal to the **autoindent** |  
40015 characters on the first line deleted shall be inserted as if entered by the user. |  
40016 Otherwise, if characters from more than one line are in the region of text: |  
40017           1. The text shall be deleted. |  
40018           2. Any text remaining in the last line in the text region shall be appended to the first line in |  
40019 the region, and the last line in the region shall be deleted. |  
40020           3. The editor shall enter text input mode after the last character not deleted from the first line |  
40021 in the text region, if any; otherwise, on the first column of the first line in the region. |  
40022 Otherwise: |  
40023           1. If the glyph for ' \$ ' is smaller than the region, the end of the region shall be marked with a |  
40024 ' \$ '. |  
40025           2. The editor shall enter text input mode, overwriting the region of text. |  
40026 *Current line/column:* As specified for the text input commands (see **Input Mode Commands in vi** |  
40027 on page 1064). |  
40028 **Change to End-of-Line** |  
40029 *Synopsis:*     [*buffer*][*count*] C |  
40030 This command shall be equivalent to the *vi* command: |  
40031 [*buffer*][*count*] c\$ |  
40032 See the **c** command. |  
40033 **Delete** |  
40034 *Synopsis:*     [*buffer*][*count*] d *motion* |  
40035 If the motion command is the **d** command repeated: |  
40036           1. The buffer text shall be in line mode. |  
40037           2. If there are less than *count* –1 lines after the current line in the edit buffer, it shall be an |  
40038 error. |  
40039           3. The text region shall be from the current line up to and including the next *count* –1 lines. |  
40040 Otherwise, the buffer text mode and text region shall be as specified by the motion command. |  
40041 If in open mode, and the current line is deleted, and the line remains on the display, an '@' |  
40042 character shall be displayed as the first glyph of that line. |



40043 Delete the region of text into *buffer*, if specified, and into the unnamed buffer. If the text to be  
 40044 deleted contains characters from more than a single line, or the buffer text is in line mode, the  
 40045 deleted text shall be copied into the numeric buffers, as well.

40046 *Current line*: Set to the first text region line that appears in the edit buffer, unless that line has  
 40047 been deleted, in which case it shall be set to the last line in the edit buffer, or line 1 if the edit  
 40048 buffer is empty.

40049 *Current column*:

- 40050 1. If the line is empty, set to column position 1.
- 40051 2. Otherwise, if the buffer text is in line mode or the motion was from the cursor toward the  
 40052 end of the edit buffer:
  - 40053 a. If a character from the current line is displayed in the current column, set to the last  
 40054 column that displays any portion of that character.
  - 40055 b. Otherwise, set to the last column in which any portion of any character in the line is  
 40056 displayed.
- 40057 3. Otherwise, if a character is displayed in the column that began the text region, set to the  
 40058 last column that displays any portion of that character.
- 40059 4. Otherwise, set to the last column in which any portion of any character in the line is  
 40060 displayed.

#### 40061 **Delete to End-of-Line**

40062 *Synopsis*: `[buffer] D`

40063 Delete the text from the current position to the end of the current line; equivalent to the *vi*  
 40064 command:

40065 `[buffer] d$`

#### 40066 **Move to End-of-Word**

40067 *Synopsis*: `[count] e`

40068 With the exception that words are used instead of bigwords as the delimiter, this command shall  
 40069 be equivalent to the *E* command.

#### 40070 **Move to End-of-Bigword**

40071 *Synopsis*: `[count] E`

40072 If the edit buffer is empty it shall be an error. If less than *count* bigwords end between the cursor  
 40073 and the end of the edit buffer, *count* shall be adjusted to the number of bigword endings between  
 40074 the cursor and the end of the edit buffer.

40075 If used as a motion command:

- 40076 1. The text region shall be from the last character of the *count*th next bigword up to and  
 40077 including the cursor character.
- 40078 2. Any text copied to a buffer shall be in character mode.

40079 If not used as a motion command:

40080 *Current line*: Set to the line containing the current column.

40081 *Current column*: Set to the last column upon which any part of the last character of the *countth*  
 40082 next bigword is displayed.

#### 40083 **Find Character in Current Line (Forward)**

40084 *Synopsis*: [ *count* ] f *character*

40085 It shall be an error if *count* occurrences of the character do not occur after the cursor in the line.

40086 If used as a motion command:

40087 1. The text range shall be from the cursor character up to and including the *countth*  
 40088 occurrence of the specified character after the cursor.

40089 2. Any text copied to a buffer shall be in character mode.

40090 If not used as a motion command:

40091 *Current line*: Unchanged.

40092 *Current column*: Set to the last column in which any portion of the *countth* occurrence of the  
 40093 specified character after the cursor appears in the line.

#### 40094 **Find Character in Current Line (Reverse)**

40095 *Synopsis*: [ *count* ] F *character*

40096 It shall be an error if *count* occurrences of the character do not occur before the cursor in the line.

40097 If used as a motion command:

40098 1. The text region shall be from the *countth* occurrence of the specified character before the  
 40099 cursor, up to, but not including the cursor character.

40100 2. Any text copied to a buffer shall be in character mode.

40101 If not used as a motion command:

40102 *Current line*: Unchanged.

40103 *Current column*: Set to the last column in which any portion of the *countth* occurrence of the  
 40104 specified character before the cursor appears in the line.

#### 40105 **Move to Line**

40106 *Synopsis*: [ *count* ] G

40107 If *count* is not specified, it shall default to the last line of the edit buffer. If *count* is greater than  
 40108 the last line of the edit buffer, it shall be an error.

40109 If used as a motion command:

40110 1. The text region shall be from the cursor line up to and including the specified line.

40111 2. Any text copied to a buffer shall be in line mode.

40112 If not used as a motion command:

40113 *Current line*: Set to *count* if *count* if specified; otherwise, the last line.

40114 *Current column*: Set to non-<blank>.

**40115 Move to Top of Screen**

40116 *Synopsis:* [count] H

40117 If the beginning of the line count greater than the first line of which any portion appears on the  
40118 display does not exist, it shall be an error.

40119 If used as a motion command:

- 40120 1. If in open mode, the text region shall be the current line.
- 40121 2. Otherwise, the text region shall be from the starting line up to and including (the first line  
40122 of the display + count -1).
- 40123 3. Any text copied to a buffer shall be in line mode.

40124 If not used as a motion command:

40125 If in open mode, this command shall set the current column to non-<blank> and do nothing else.

40126 Otherwise, it shall set the current line and current column as follows.

40127 *Current line:* Set to (the first line of the display + count -1).

40128 *Current column:* Set to non-<blank>.

**40129 Insert Before Cursor**

40130 *Synopsis:* [count] i

40131 Enter text input mode before the current cursor position. No characters already in the edit buffer  
40132 shall be affected by this command. A *count* shall cause the input text to be appended *count* -1  
40133 more times to the end of the input.

40134 *Current line/column:* As specified for the text input commands (see **Input Mode Commands in vi**  
40135 on page 1064).

**40136 Insert at Beginning of Line**

40137 *Synopsis:* [count] I

40138 This command shall be equivalent to the *vi* command `^[count]i` command.

**40139 Join**

40140 *Synopsis:* [count] J

40141 If the current line is the last line in the edit buffer, it shall be an error.

40142 This command shall be equivalent to the *ex join* command with no addresses, and an *ex*  
40143 command *count* value of 1 if *count* was not specified or if a *count* of 1 was specified, and an *ex*  
40144 command *count* value of *count* -1 for any other value of *count*, except that the current line and  
40145 column shall be set as follows.

40146 *Current line:* Unchanged.

40147 *Current column:* The last column in which any portion of the character following the last  
40148 character in the initial line is displayed, or the last character in the line if no characters were  
40149 appended.

## 40150 **Move to Bottom of Screen**

40151 *Synopsis:*     [ *count* ] L

40152 If the beginning of the line count less than the last line of which any portion appears on the  
40153 display does not exist, it shall be an error.

40154 If used as a motion command:

- 40155     1. If in open mode, the text region shall be the current line.
- 40156     2. Otherwise, the text region shall include all lines from the starting cursor line to (the last  
40157 line of the display  $-(count - 1)$ ).
- 40158     3. Any text copied to a buffer shall be in line mode.

40159 If not used as a motion command:

- 40160     1. If in open mode, this command shall set the current column to non-<blank> and do  
40161 nothing else.
- 40162     2. Otherwise, it shall set the current line and current column as follows.

40163 *Current line:* Set to (the last line of the display  $-(count - 1)$ ).

40164 *Current column:* Set to non-<blank>.

## 40165 **Mark Position**

40166 *Synopsis:*     m *letter*

40167 This command shall be equivalent to the *ex mark* command with the specified character as an  
40168 argument.

## 40169 **Move to Middle of Screen**

40170 *Synopsis:*     M

40171 The middle line of the display shall be calculated as follows:

40172  $(\text{the top line of the display}) + (((\text{number of lines displayed}) + 1) / 2) - 1$

40173 If used as a motion command:

- 40174     1. If in open mode, the text region shall be the current line.
- 40175     2. Otherwise, the text region shall include all lines from the starting cursor line up to and  
40176 including the middle line of the display.
- 40177     3. Any text copied to a buffer shall be in line mode.

40178 If not used as a motion command:

40179 If in open mode, this command shall set the current column to non-<blank> and do nothing else.

40180 Otherwise, it shall set the current line and current column as follows.

40181 *Current line:* Set to the middle line of the display.

40182 *Current column:* Set to non-<blank>.

### 40183 **Repeat Regular Expression Find (Forward)**

40184 *Synopsis:*     n

40185 If the remembered search direction was forward, the **n** command shall be equivalent to the *vi /*  
40186 command with no characters entered by the user. Otherwise, it shall be equivalent to the *vi ?*  
40187 command with no characters entered by the user.

40188 If the **n** command is used as a motion command for the **!** command, the editor shall not enter  
40189 text input mode on the last line on the screen, and shall behave as if the user entered a single '!'  
40190 character as the text input.

### 40191 **Repeat Regular Expression Find (Reverse)**

40192 *Synopsis:*     N

40193 Scan for the next match of the last pattern given to */* or *?*, but in the reverse direction; this is the  
40194 reverse of **n**.

40195 If the remembered search direction was forward, the **N** command shall be equivalent to the *vi ?*  
40196 command with no characters entered by the user. Otherwise, it shall be equivalent to the *vi /*  
40197 command with no characters entered by the user. If the **N** command is used as a motion  
40198 command for the **!** command, the editor shall not enter text input mode on the last line on the  
40199 screen, and shall behave as if the user entered a single **!** character as the text input.

### 40200 **Insert Empty Line Below**

40201 *Synopsis:*     o

40202 Enter text input mode in a new line appended after the current line. A *count* shall cause the input  
40203 text to be appended *count* – 1 more times to the end of the already added text, each time starting  
40204 on a new, appended line.

40205 *Current line/column:* As specified for the text input commands (see **Input Mode Commands in vi**  
40206 on page 1064).

### 40207 **Insert Empty Line Above**

40208 *Synopsis:*     O

40209 Enter text input mode in a new line inserted before the current line. A *count* shall cause the input  
40210 text to be appended *count* – 1 more times to the end of the already added text, each time starting  
40211 on a new, appended line.

40212 *Current line/column:* As specified for the text input commands (see **Input Mode Commands in vi**  
40213 on page 1064).

### 40214 **Put from Buffer Following**

40215 *Synopsis:*     [*buffer*] p

40216 If no *buffer* is specified, the unnamed buffer shall be used.

40217 If the buffer text is in line mode, the text shall be appended below the current line, and each line  
40218 of the buffer shall become a new line in the edit buffer. A *count* shall cause the buffer text to be  
40219 appended *count* – 1 more times to the end of the already added text, each time starting on a new,  
40220 appended line.

40221 If the buffer text is in character mode, the text shall be appended into the current line after the  
40222 cursor, and each line of the buffer other than the first and last shall become a new line in the edit

40223 buffer. A *count* shall cause the buffer text to be appended *count* –1 more times to the end of the  
 40224 already added text, each time starting after the last added character.

40225 *Current line*: If the buffer text is in line mode, set the line to line +1; otherwise, unchanged.

40226 *Current column*: If the buffer text is in line mode:

40227 1. If there is a non-<blank> character in the first line of the buffer, set to the last column on  
 40228 which any portion of the first non-<blank> character in the line is displayed.

40229 2. If there is no non-<blank> character in the first line of the buffer, set to the last column on  
 40230 which any portion of the last character in the first line of the buffer is displayed.

40231 If the buffer text is in character mode:

40232 1. If the text in the buffer is from more than a single line, then set to the last column on which  
 40233 any portion of the first character from the buffer is displayed.

40234 2. Otherwise, if the buffer is the unnamed buffer, set to the last column on which any portion  
 40235 of the last character from the buffer is displayed.

40236 3. Otherwise, set to the first column on which any portion of the first character from the  
 40237 buffer is displayed.

#### 40238 **Put from Buffer Before**

40239 *Synopsis*: [ *buffer* ] P

40240 If no *buffer* is specified, the unnamed buffer shall be used.

40241 If the buffer text is in line mode, the text shall be inserted above the current line, and each line of  
 40242 the buffer shall become a new line in the edit buffer. A *count* shall cause the buffer text to be  
 40243 appended *count* –1 more times to the end of the already added text, each time starting on a new,  
 40244 appended line.

40245 If the buffer text is in character mode, the text shall be inserted into the current line before the  
 40246 cursor, and each line of the buffer other than the first and last shall become a new line in the edit  
 40247 buffer. A *count* shall cause the buffer text to be appended *count* –1 more times to the end of the  
 40248 already added text, each time starting after the last added character.

40249 *Current line*: Unchanged.

40250 *Current column*: If the buffer text is in line mode:

40251 1. If there is a non-<blank> character in the first line of the buffer, set to the last column on  
 40252 which any portion of that character is displayed.

40253 2. If there is no non-<blank> character in the first line of the buffer, set to the last column on  
 40254 which any portion of the last character in the first line of the buffer is displayed.

40255 If the buffer text is in character mode:

40256 1. If the buffer is the unnamed buffer, set to the last column on which any portion of the last  
 40257 character from the buffer is displayed.

40258 2. Otherwise, set to the first column on which any portion of the first character from the  
 40259 buffer is displayed.

40260      **Enter ex Mode**40261      *Synopsis:*      Q40262      Leave visual or open mode and enter *ex* command mode.40263      *Current line:* Unchanged.40264      *Current column:* Unchanged.40265      **Replace Character**40266      *Synopsis:*      [count] r character40267      Replace the *count* characters at and after the cursor with the specified character. If there are less  
40268      than *count* characters at and after the cursor on the line, it shall be an error.40269      If character is <control>-V, any next character other than the <newline> shall be stripped of any  
40270      special meaning and used as a literal character.40271      If character is <ESC>, no replacement shall be made and the current line and current column  
40272      shall be unchanged.40273      If character is <carriage-return> or <newline>, *count* new lines shall be appended to the current  
40274      line. All but the last of these lines shall be empty. *count* characters at and after the cursor shall be  
40275      discarded, and any remaining characters after the cursor in the current line shall be moved to the  
40276      last of the new lines. If the **autoindent** edit option is set, they shall be preceded by the same  
40277      number of **autoindent** characters found on the line from which the command was executed.40278      *Current line:* Unchanged unless the replacement character is a <carriage-return> or <newline>  
40279      character, in which case it shall be set to line + *count*.40280      *Current column:* Set to the last column position on which a portion of the last replaced character  
40281      is displayed, or if the replacement character caused new lines to be created, set to non-<blank>.40282      **Replace Characters**40283      *Synopsis:*      R40284      Enter text input mode at the current cursor position. A *count* shall cause the input text to be  
40285      appended *count* - 1 more times to the end of the input.40286      *Current line/column:* As specified for the text input commands (see **Input Mode Commands in vi**  
40287      on page 1064).40288      **Substitute Character**40289      *Synopsis:*      [buffer][count] s40290      This command shall be equivalent to the *vi* command:

40291      [buffer][count] c&lt;space&gt;

## 40292 **Substitute Lines**

40293 *Synopsis:*     [*buffer*][*count*] S

40294 This command shall be equivalent to the *vi* command:

40295 [*buffer*][*count*] c\_

## 40296 **Move Cursor to Before Character (Forward)**

40297 *Synopsis:*     [*count*] t *character*

40298 It shall be an error if *count* occurrences of the character do not occur after the cursor in the line.

40299 If used as a motion command:

40300     1. The text region shall be from the cursor up to but not including the *count*th occurrence of  
40301       the specified character after the cursor.

40302     2. Any text copied to a buffer shall be in character mode.

40303 If not used as a motion command:

40304 *Current line:* Unchanged.

40305 *Current column:* Set to the last column in which any portion of the character before the *count*th  
40306 occurrence of the specified character after the cursor appears in the line.

## 40307 **Move Cursor to After Character (Reverse)**

40308 *Synopsis:*     [*count*] T *character*

40309 It shall be an error if *count* occurrences of the character do not occur before the cursor in the line.

40310 If used as a motion command:

40311     1. If the character before the cursor is the specified character, it shall be an error.

40312     2. The text region shall be from the character before the cursor up to but not including the  
40313       *count*th occurrence of the specified character before the cursor.

40314     3. Any text copied to a buffer shall be in character mode.

40315 If not used as a motion command:

40316 *Current line:* Unchanged.

40317 *Current column:* Set to the last column in which any portion of the character after the *count*th  
40318 occurrence of the specified character before the cursor appears in the line.

## 40319 **Undo**

40320 *Synopsis:*     u

40321 This command shall be equivalent to the ex **undo** command except that the current line and  
40322 current column shall be set as follows:

40323 *Current line:* Set to the first line added or changed if any; otherwise, move to the line preceding  
40324 any deleted text if one exists; otherwise, move to line 1.

40325 *Current column:* If undoing an ex command, set to the first non-<blank> character.

40326 Otherwise, if undoing a text input command:



40327 1. If the command was an **C**, **c**, **O**, **o**, **R**, **S**, or **s** command, the current column shall be set to  
40328 the value it held when the text input command was entered.

40329 2. Otherwise, set to the last column in which any portion of the first character after the  
40330 deleted text is displayed, or, if no characters follow the text deleted from this line, set to the  
40331 last column in which any portion of the last character in the line is displayed, or 1 if the line  
40332 is empty.

40333 Otherwise, if a single line was modified (that is, not added or deleted) by the **u** command:

40334 1. If text was added or changed, set to the last column in which any portion of the first  
40335 character added or changed is displayed.

40336 2. If text was deleted, set to the last column in which any portion of the first character after  
40337 the deleted text is displayed, or, if no characters follow the deleted text, set to the last  
40338 column in which any portion of the last character in the line is displayed, or 1 if the line is  
40339 empty.

40340 Otherwise, set to non-<blank>.

#### 40341 **Undo Current Line**

40342 *Synopsis:*     **U**

40343 Restore the current line to its state immediately before the most recent time that it became the  
40344 current line.

40345 *Current line:* Unchanged.

40346 *Current column:* Set to the first column in the line in which any portion of the first character in  
40347 the line is displayed.

#### 40348 **Move to Beginning of Word**

40349 *Synopsis:*     [*count*] **w**

40350 With the exception that words are used as the delimiter instead of bigwords, this command shall  
40351 be equivalent to the **W** command.

#### 40352 **Move to Beginning of Bigword**

40353 *Synopsis:*     [*count*] **W**

40354 If the edit buffer is empty, it shall be an error. If there are less than *count* bigwords between the  
40355 cursor and the end of the edit buffer, *count* shall be adjusted to move the cursor to the last  
40356 bigword in the edit buffer.

40357 If used as a motion command:

40358 1. If the associated command is **c**, *count* is 1, and the cursor is on a <blank> character, the  
40359 region of text shall be the current character and no further action shall be taken.

40360 2. If there are less than *count* bigwords between the cursor and the end of the edit buffer, then  
40361 the command shall succeed, and the region of text shall include the last character of the  
40362 edit buffer.

40363 3. If there are <blank> characters or an end-of-line that precede the *count*th bigword, and the  
40364 associated command is **c**, the region of text shall be up to and including the last character  
40365 before the preceding <blank> characters or end-of-line.

40366 4. If there are <blank> characters or an end-of-line that precede the bigword, and the  
 40367 associated command is **d** or **y**, the region of text shall be up to and including the last  
 40368 <blank> character before the start of the bigword or end-of-line.

40369 5. Any text copied to a buffer shall be in character mode.

40370 If not used as a motion command:

40371 1. If the cursor is on the last character of the edit buffer, it shall be an error.

40372 *Current line*: Set to the line containing the current column.

40373 *Current column*: Set to the last column in which any part of the first character of the *count*th next  
 40374 bigword is displayed.

### 40375 **Delete Character at Cursor**

40376 *Synopsis*: `[buffer][count] x`

40377 Delete the *count* characters at and after the current character into *buffer*, if specified, and into the  
 40378 unnamed buffer.

40379 If the line is empty, it shall be an error. If there are less than *count* characters at and after the  
 40380 cursor on the current line, *count* shall be adjusted to the number of characters at and after the  
 40381 cursor.

40382 *Current line*: Unchanged.

40383 *Current column*: If the line is empty, set to column position 1. Otherwise, if there were *count* or  
 40384 less characters at and after the cursor on the current line, set to the last column that displays any  
 40385 part of the last character of the line. Otherwise, unchanged.

### 40386 **Delete Character Before Cursor**

40387 *Synopsis*: `[buffer][count] X`

40388 Delete the *count* characters before the current character into *buffer*, if specified, and into the  
 40389 unnamed buffer.

40390 If there are no characters before the current character on the current line, it shall be an error. If  
 40391 there are less than *count* previous characters on the current line, *count* shall be adjusted to the  
 40392 number of previous characters on the line.

40393 *Current line*: Unchanged.

40394 *Current column*: Set to (*current column* – *the width of the deleted characters*).

### 40395 **Yank**

40396 *Synopsis*: `[buffer][count] y motion`

40397 Copy (yank) the region of text into *buffer*, if specified, and into the unnamed buffer.

40398 If the motion command is the **y** command repeated:

40399 1. The buffer shall be in line mode.

40400 2. If there are less than *count* – 1 lines after the current line in the edit buffer, it shall be an  
 40401 error.

40402 3. The text region shall be from the current line up to and including the next *count* – 1 lines.

|       |                                                                                                                             |  |
|-------|-----------------------------------------------------------------------------------------------------------------------------|--|
| 40403 | Otherwise, the buffer text mode and text region shall be as specified by the motion command.                                |  |
| 40404 | <i>Current line:</i> If the motion was from the current cursor position toward the end of the edit                          |  |
| 40405 | buffer, unchanged. Otherwise, set to the first line in the edit buffer that is part of the text region                      |  |
| 40406 | specified by the motion command.                                                                                            |  |
| 40407 | <i>Current column:</i>                                                                                                      |  |
| 40408 | 1. If the motion was from the current cursor position toward the end of the edit buffer,                                    |  |
| 40409 | unchanged.                                                                                                                  |  |
| 40410 | 2. Otherwise, if the current line is empty, set to column position 1.                                                       |  |
| 40411 | 3. Otherwise, set to the last column that displays any part of the first character in the file that                         |  |
| 40412 | is part of the text region specified by the motion command.                                                                 |  |
| 40413 | <b>Yank Current Line</b>                                                                                                    |  |
| 40414 | <i>Synopsis:</i> [ <i>buffer</i> ][ <i>count</i> ] Y                                                                        |  |
| 40415 | This command shall be equivalent to the <i>vi</i> command:                                                                  |  |
| 40416 | [ <i>buffer</i> ][ <i>count</i> ] y_                                                                                        |  |
| 40417 | <b>Redraw Window</b>                                                                                                        |  |
| 40418 | If in open mode, the <i>z</i> command shall have the Synopsis:                                                              |  |
| 40419 | <i>Synopsis:</i> [ <i>count</i> ] z                                                                                         |  |
| 40420 | If <i>count</i> is not specified, it shall default to the <b>window</b> edit option -1. The <i>z</i> command shall be       |  |
| 40421 | equivalent to the <i>ex z</i> command, with a type character of = and a <i>count</i> of <i>count</i> -2, except that        |  |
| 40422 | the current line and current column shall be set as follows, and the <b>window</b> edit option shall not                    |  |
| 40423 | be affected. If the calculation for the <i>count</i> argument would result in a negative number, the                        |  |
| 40424 | <i>count</i> argument to the <i>ex z</i> command shall be zero. A blank line shall be written after the last line           |  |
| 40425 | is written.                                                                                                                 |  |
| 40426 | <i>Current line:</i> Unchanged.                                                                                             |  |
| 40427 | <i>Current column:</i> Unchanged.                                                                                           |  |
| 40428 | If not in open mode, the <i>z</i> command shall have the following Synopsis:                                                |  |
| 40429 | <i>Synopsis:</i> [ <i>line</i> ] z [ <i>count</i> ] <i>character</i>                                                        |  |
| 40430 | If <i>line</i> is not specified, it shall default to the current line. If <i>line</i> is specified, but is greater than the |  |
| 40431 | number of lines in the edit buffer, it shall default to the number of lines in the edit buffer.                             |  |
| 40432 | If <i>count</i> is specified, the value of the <b>window</b> edit option shall be set to <i>count</i> (as described in the  |  |
| 40433 | <i>ex window</i> command), and the screen shall be redrawn.                                                                 |  |
| 40434 | <i>line</i> shall be placed as specified by the following characters:                                                       |  |
| 40435 | <newline>, <carriage-return>                                                                                                |  |
| 40436 | Place the beginning of the line on the first line of the display.                                                           |  |
| 40437 | . Place the beginning of the line in the center of the display. The middle line of the display                              |  |
| 40438 | shall be calculated as described for the <b>M</b> command.                                                                  |  |
| 40439 | - Place an unspecified portion of the line on the last line of the display.                                                 |  |
| 40440 | + If <i>line</i> was specified, equivalent to the <newline> case. If <i>line</i> was not specified, display a               |  |
| 40441 | screen where the first line of the display shall be (current last line) +1. If there are no lines                           |  |

40442 after the last line in the display, it shall be an error.

40443 ^ If *line* was specified, display a screen where the last line of the display shall contain an  
 40444 unspecified portion of the first line of a display that had an unspecified portion of the  
 40445 specified line on the last line of the display. If this calculation results in a line before the  
 40446 beginning of the edit buffer, display the first screen of the edit buffer.

40447 Otherwise, display a screen where the last line of the display shall contain an unspecified  
 40448 portion of (current first line -1). If this calculation results in a line before the beginning of  
 40449 the edit buffer, it shall be an error.

40450 *Current line:* If *line* and the ' ^ ' character were specified:

40451 1. If the first screen was displayed as a result of the command attempting to display lines  
 40452 before the beginning of the edit buffer: if the first screen was already displayed,  
 40453 unchanged; otherwise, set to (current first line -1).

40454 2. Otherwise, set to the last line of the display.

40455 If *line* and the ' + ' character were specified, set to the first line of the display.

40456 Otherwise, if *line* was specified, set to *line*.

40457 Otherwise, unchanged.

40458 *Current column:* Set to non-<blank>.

40459 **Exit**

40460 *Synopsis:*      ZZ

40461 This command shall be equivalent to the **ex xit** command with no addresses, trailing !, or file  
 40462 name (see the **ex xit** command).

40463 **Input Mode Commands in vi**

40464 In text input mode, the current line shall consist of zero or more of the following categories:

40465 1. Characters preceding the text input entry point  
 40466 Characters in this category shall not be modified during text input mode.

40467 2. **autoindent** characters

40468 **autoindent** characters shall be automatically inserted into each line that is created in text  
 40469 input mode, either as a result of entering a <newline> character or <carriage-return>  
 40470 character while in text input mode, or as an effect of the command itself; for example, **O** or  
 40471 **o** (see the **ex autoindent** command), as if entered by the user.

40472 It shall be possible to erase **autoindent** characters with the <control>-D command; it is  
 40473 unspecified whether they can be erased by <control>-H, <control>-U, and <control>-W  
 40474 characters. Erasing any **autoindent** character turns the glyph into erase-columns and  
 40475 deletes the character from the edit buffer, but does not change its representation on the  
 40476 screen.

40477 3. Text input characters

40478 Text input characters are the characters entered by the user. Erasing any text input  
 40479 character turns the glyph into erase-columns and deletes the character from the edit buffer,  
 40480 but does not change its representation on the screen.

- Each text input character entered by the user (that does not have a special meaning) shall be treated as follows:
- a. The text input character shall be appended to the last character in the edit buffer from the first, second, or third categories.
  - b. If there are no erase-columns on the screen, the text input command was the **R** command, and characters in the fifth category from the original line follow the cursor, the next such character shall be deleted from the edit buffer. If the **slowopen** edit option is not set, the corresponding glyph on the screen shall become erase-columns.
  - c. If there are erase-columns on the screen, as many columns as they occupy, or as are necessary, shall be overwritten to display the text input character. (If only part of a multi-column glyph is overwritten, the remainder shall be left on the screen, and continue to be treated as erase-columns; it is unspecified whether the remainder of the glyph is modified in any way.)
  - d. If additional screen columns are needed to display the text input character:
    1. If the **slowopen** edit option is set, the text input characters shall be displayed on subsequent screen columns, overwriting any characters displayed in those columns.
    2. Otherwise, any characters currently displayed on or after the column on the screen where the text input character is to be displayed shall be pushed ahead the number of screen columns necessary to display the rest of the text input character.
4. Erase-columns
- Erase-columns are not logically part of the edit buffer, appearing only on the screen, and may be overwritten on the screen by subsequent text input characters. When text input mode ends, all erase-columns shall no longer appear on the screen.
- Erase-columns are initially the region of text specified by the **c** command ( see **Change** on page 1051) however, erasing **autoindent** or text input characters causes the glyphs of the erased characters to be treated as erase-columns.
5. Characters following the text region for the **c** command, or the text input entry point for all other commands
- Characters in this category shall not be modified during text input mode, except as specified in category 3.b. for the **R** text input command, or as <blank> characters deleted when a <newline> character or <carriage-return> character is entered.
- It is unspecified whether it is an error to attempt to erase past the beginning of a line that was created by the entry of a <newline> or <carriage-return> character during text input mode. If it is not an error, the editor shall behave as if the erasing character was entered immediately after the last text input character entered on the previous line, and all of the characters on the current line shall be treated as erase-columns.
- When text input mode is entered, or after a text input mode character is entered (except as specified for the special characters below), the cursor shall be positioned as follows:
1. On the first column that displays any part of the first erase-column, if one exists
  2. Otherwise, if the **slowopen** edit option is set, on the first screen column after the last character in the first, second, or third categories, if one exists

- 40525 3. Otherwise, the first column that displays any part of the first character in the fifth category,  
40526 if one exists
- 40527 4. Otherwise, the screen column after the last character in the first, second, or third  
40528 categories, if one exists
- 40529 5. Otherwise, on column position 1

40530 The characters that are updated on the screen during text input mode are unspecified, other than  
40531 that the last text input character shall always be updated, and, if the **slowopen** edit option is not  
40532 set, the current cursor character shall always be updated.

40533 The following specifications are for command characters entered during text input mode.

#### 40534 NUL

40535 *Synopsis:* NUL

40536 If the first character of the text input is a NUL, the most recently input text shall be input as if  
40537 entered by the user, and then text input mode shall be exited. The text shall be input literally;  
40538 that is, characters are neither macro or abbreviation expanded, nor are any characters interpreted  
40539 in any special manner. It is unspecified whether implementations shall support more than 256  
40540 bytes of remembered input text.

#### 40541 <control>-D

40542 *Synopsis:* <control>-D

40543 The <control>-D character shall have no special meaning when in text input mode for a line-  
40544 oriented command (see **Command Descriptions in vi** on page 1031).

40545 This command need not be supported on block-mode terminals.

40546 If the cursor does not follow an **autoindent** character, or an **autoindent** character and a '0' or  
40547 '^' character:

- 40548 1. If the cursor is in column position 1, the <control>-D character shall be discarded and no  
40549 further action taken.
- 40550 2. Otherwise, the <control>-D character shall have no special meaning.

40551 If the last input character was a '0', the cursor shall be moved to column position 1.

40552 Otherwise, if the last input character was a '^', the cursor shall be moved to column position 1.  
40553 In addition, the **autoindent** level for the next input line shall be derived from the same line from  
40554 which the **autoindent** level for the current input line was derived.

40555 Otherwise, the cursor shall be moved back to the column after the previous shiftwidth (see the  
40556 **ex shiftwidth** command) boundary.

40557 All of the glyphs on columns between the starting cursor position and (inclusively) the ending  
40558 cursor position shall become erase-columns as described in **Input Mode Commands in vi** on  
40559 page 1064.

40560 *Current line:* Unchanged.

40561 *Current column:* Set to 1 if the <control>-D was preceded by a '^' or '0'; otherwise, set to  
40562 (column - 1) - ((column - 2) % **shiftwidth**).

|       |                                                                                                                 |                                |
|-------|-----------------------------------------------------------------------------------------------------------------|--------------------------------|
| 40563 | <b>&lt;control&gt;-H</b>                                                                                        |                                |
| 40564 | <i>Synopsis:</i>                                                                                                | <b>&lt;control&gt;-H</b>       |
| 40565 | If in text input mode for a line-oriented command, and there are no characters to erase, text                   |                                |
| 40566 | input mode shall be terminated, no further action shall be done for this command, and the                       |                                |
| 40567 | current line and column shall be unchanged.                                                                     |                                |
| 40568 | If there are characters other than <b>autoindent</b> characters that have been input on the current line        |                                |
| 40569 | before the cursor, the cursor shall move back one character.                                                    |                                |
| 40570 | Otherwise, if there are <b>autoindent</b> characters on the current line before the cursor, it is               |                                |
| 40571 | implementation-dependent whether the <b>&lt;control&gt;-H</b> command is an error or if the cursor              |                                |
| 40572 | moves back one <b>autoindent</b> character.                                                                     |                                |
| 40573 | Otherwise, if the cursor is in column position 1 and there are previous lines that have been input,             |                                |
| 40574 | it is implementation-dependent whether the <b>&lt;control&gt;-H</b> command is an error or if it is             |                                |
| 40575 | equivalent to entering <b>&lt;control&gt;-H</b> after the last input character on the previous input line.      |                                |
| 40576 | Otherwise, it shall be an error.                                                                                |                                |
| 40577 | All of the glyphs on columns between the starting cursor position and (inclusively) the ending                  |                                |
| 40578 | cursor position shall become erase-columns as described in <b>Input Mode Commands in vi</b> on                  |                                |
| 40579 | page 1064.                                                                                                      |                                |
| 40580 | The current erase character (see <i>stty</i> ) shall cause an equivalent action to the <b>&lt;control&gt;-H</b> |                                |
| 40581 | command, unless the previously inserted character was a backslash, in which case it shall be as                 |                                |
| 40582 | if the literal current erase character had been inserted instead of the backslash.                              |                                |
| 40583 | <i>Current line:</i> Unchanged, unless previously input lines are erased, in which case it shall be set to      |                                |
| 40584 | line -1.                                                                                                        |                                |
| 40585 | <i>Current column:</i> Set to the first column that displays any portion of the character backed up             |                                |
| 40586 | over.                                                                                                           |                                |
| 40587 | <b>&lt;newline&gt;</b>                                                                                          |                                |
| 40588 | <i>Synopsis:</i>                                                                                                | <b>&lt;newline&gt;</b>         |
| 40589 |                                                                                                                 | <b>&lt;carriage-return&gt;</b> |
| 40590 |                                                                                                                 | <b>&lt;control&gt;-J</b>       |
| 40591 |                                                                                                                 | <b>&lt;control&gt;-M</b>       |
| 40592 | If input was part of a line-oriented command, text input mode shall be terminated and the                       |                                |
| 40593 | command shall continue execution with the input provided.                                                       |                                |
| 40594 | Otherwise, terminate the current line. If there are no characters other than <b>autoindent</b> characters       |                                |
| 40595 | on the line, all characters on the line shall be discarded. Otherwise, it is unspecified whether the            |                                |
| 40596 | <b>autoindent</b> characters in the line are modified by entering these characters.                             |                                |
| 40597 | Continue text input mode on a new line appended after the current line. If the <b>slowopen</b> edit             |                                |
| 40598 | option is set, the lines on the screen below the current line shall not be pushed down, but the                 |                                |
| 40599 | first of them shall be cleared and shall appear to be overwritten. Otherwise, the lines of the                  |                                |
| 40600 | screen below the current line shall be pushed down.                                                             |                                |
| 40601 | If the <b>autoindent</b> edit option is set, an appropriate number of <b>autoindent</b> characters shall be     |                                |
| 40602 | added as a prefix to the line as described by the <b>ex autoindent</b> edit option.                             |                                |
| 40603 | All columns after the cursor that are erase-columns (as described in <b>Input Mode Commands in</b>              |                                |
| 40604 | <b>vi</b> on page 1064) shall be discarded.                                                                     |                                |

40605 If the **autoindent** edit option is set, all <blank> characters immediately following the cursor shall  
 40606 be discarded.

40607 All remaining characters after the cursor shall be transferred to the new line, positioned after any  
 40608 **autoindent** characters.

40609 *Current line:* Set to current line +1.

40610 *Current column:* Set to the first column that displays any portion of the first character after the  
 40611 **autoindent** characters on the new line, if any, or the first column position after the last  
 40612 **autoindent** character, if any, or column position 1.

40613 **<control>-T**

40614 *Synopsis:*     <control>-T

40615 The <control>-T character shall have no special meaning when in text input mode for a line-  
 40616 oriented command (see **Command Descriptions in vi** on page 1031).

40617 This command need not be supported on block-mode terminals.

40618 Behave as if the user entered the minimum number of <<blank>> characters necessary to move  
 40619 the cursor forward to the column position after the next **shiftwidth** (see the *ex* **shiftwidth**  
 40620 command) boundary.

40621 *Current line:* Unchanged.

40622 *Current column:* Set to  $column + shiftwidth - ((column - 1) \% shiftwidth)$ .

40623 **<control>-U**

40624 *Synopsis:*     <control>-U

40625 If there are characters other than **autoindent** characters that have been input on the current line  
 40626 before the cursor, the cursor shall move to the first character input after the **autoindent**  
 40627 characters.

40628 Otherwise, if there are **autoindent** characters on the current line before the cursor, it is  
 40629 implementation-dependent whether the <control>-U command is an error or if the cursor moves  
 40630 to the first column position on the line.

40631 Otherwise, if the cursor is in column position 1 and there are previous lines that have been input,  
 40632 it is implementation-dependent whether the <control>-U command is an error or if it is  
 40633 equivalent to entering <control>-U after the last input character on the previous input line.

40634 Otherwise, it shall be an error.

40635 All of the glyphs on columns between the starting cursor position and (inclusively) the ending  
 40636 cursor position shall become erase-columns as described in **Input Mode Commands in vi** on  
 40637 page 1064.

40638 The current *kill* character (see *stty*) shall cause an equivalent action to the <control>-U  
 40639 command, unless the previously inserted character was a backslash, in which case it shall be as  
 40640 if the literal current *kill* character had been inserted instead of the backslash.

40641 *Current line:* Unchanged, unless previously input lines are erased, in which case it shall be set to  
 40642 line -1.

40643 *Current column:* Set to the first column that displays any portion of the last character backed up  
 40644 over.



40645        **<control>-V**

40646        *Synopsis:*        <control>-V  
 40647                    <control>-Q

40648        Allow the entry of any subsequent character, other than <control>-J or the character, as a literal  
 40649        character, removing any special meaning that it may have to the editor in text input mode. If a  
 40650        <control>-V or <control>-Q is entered before a <control>-J or <newline> character, the  
 40651        <control>-V or <control>-Q character shall be discarded, and the <control>-J or <newline> shall  
 40652        behave as described in the <newline> command character during input mode.

40653        For purposes of the display only, the editor shall behave as if a '^' character was entered, and  
 40654        the cursor shall be positioned as if overwriting the '^' character. When a subsequent character  
 40655        is entered, the editor shall behave as if that character was entered instead of the original  
 40656        <control>-V or <control>-Q character.

40657        *Current line:* Unchanged.

40658        *Current column:* Unchanged.

40659        **<control>-W**

40660        *Synopsis:*        <control>-W

40661        If there are characters other than **autoindent** characters that have been input on the current line  
 40662        before the cursor, the cursor shall move back over the last word preceding the cursor (including  
 40663        any <blank> characters between the end of the last word and the current cursor); the cursor shall  
 40664        not move to before the first character after the end of any **autoindent** characters.

40665        Otherwise, if there are **autoindent** characters on the current line before the cursor, it is  
 40666        implementation-dependent whether the <control>-W command is an error or if the cursor  
 40667        moves to the first column position on the line.

40668        Otherwise, if the cursor is in column position 1 and there are previous lines that have been input,  
 40669        it is implementation-dependent whether the <control>-W command is an error or if it is  
 40670        equivalent to entering <control>-W after the last input character on the previous input line.

40671        Otherwise, it shall be an error.

40672        All of the glyphs on columns between the starting cursor position and (inclusively) the ending  
 40673        cursor position shall become erase-columns as described in **Input Mode Commands in vi** on  
 40674        page 1064.

40675        *Current line:* Unchanged, unless previously input lines are erased, in which case it shall be set to  
 40676        line -1.

40677        *Current column:* Set to the first column that displays any portion of the last character backed up  
 40678        over.

40679        **<ESC>**

40680        *Synopsis:*        <ESC>

40681        If input was part of a line-oriented command:

- 40682        1. If *interrupt* was entered, text input mode shall be terminated and the editor shall return to  
 40683        command mode. The terminal shall be alerted.
- 40684        2. If <ESC> was entered, text input mode shall be terminated and the command shall  
 40685        continue execution with the input provided.

|       |                                                                                                                        |
|-------|------------------------------------------------------------------------------------------------------------------------|
| 40686 | Otherwise, terminate text input mode and return to command mode.                                                       |
| 40687 | Any <b>autoindent</b> characters entered on newly created lines that have no other characters shall be                 |
| 40688 | deleted.                                                                                                               |
| 40689 | Any leading <b>autoindent</b> and <<blank>> characters on newly created lines shall be rewritten to                    |
| 40690 | be the minimum number of <<blank>> characters possible.                                                                |
| 40691 | The screen shall be redisplayed as necessary to match the contents of the edit buffer.                                 |
| 40692 | <i>Current line:</i> Unchanged.                                                                                        |
| 40693 | <i>Current column:</i>                                                                                                 |
| 40694 | 1. If there are text input characters on the current line, the column shall be set to the last                         |
| 40695 | column where any portion of the last text input character is displayed.                                                |
| 40696 | 2. Otherwise, if a character is displayed in the current column, unchanged.                                            |
| 40697 | 3. Otherwise, set to column position 1.                                                                                |
| 40698 | <b>EXIT STATUS</b>                                                                                                     |
| 40699 | The following exit values shall be returned:                                                                           |
| 40700 | 0 Successful completion.                                                                                               |
| 40701 | >0 An error occurred.                                                                                                  |
| 40702 | <b>CONSEQUENCES OF ERRORS</b>                                                                                          |
| 40703 | When any error is encountered and the standard input is not a terminal device file, <i>vi</i> shall not                |
| 40704 | write the file or return to command or text input mode, and shall terminate with a non-zero exit                       |
| 40705 | status.                                                                                                                |
| 40706 | Otherwise, when an unrecoverable error is encountered it shall be equivalent to a SIGHUP                               |
| 40707 | asynchronous event.                                                                                                    |
| 40708 | Otherwise, when an error is encountered, the editor shall behave as specified in <b>Command</b>                        |
| 40709 | <b>Descriptions in vi</b> on page 1031.                                                                                |
| 40710 | <b>APPLICATION USAGE</b>                                                                                               |
| 40711 | Application writers should note that this utility need not be provided on systems that do not                          |
| 40712 | support the User Portability Utilities option.                                                                         |
| 40713 | <b>EXAMPLES</b>                                                                                                        |
| 40714 | None.                                                                                                                  |
| 40715 | <b>RATIONALE</b>                                                                                                       |
| 40716 | See the RATIONALE for <i>ex</i> for more information on <i>vi</i> . Major portions of the <i>vi</i> utility            |
| 40717 | specification point to <i>ex</i> to avoid inadvertent divergence. While <i>ex</i> and <i>vi</i> have historically been |
| 40718 | implemented as a single utility, this is not required by IEEE Std. 1003.1-200x.                                        |
| 40719 | It is recognized that portions of <i>vi</i> would be difficult, if not impossible, to implement                        |
| 40720 | satisfactorily on a block-mode terminal, or a terminal without any form of cursor addressing,                          |
| 40721 | thus it is not a mandatory requirement that such features should work on all terminals. It is the                      |
| 40722 | intention, however, that a <i>vi</i> implementation should provide the full set of capabilities on all                 |
| 40723 | terminals capable of supporting them.                                                                                  |
| 40724 | Historically, <i>vi</i> exited immediately if the standard input was not a terminal.                                   |
| 40725 | IEEE Std. 1003.1-200x permits, but does not require, this behavior. An end-of-file condition is not                    |
| 40726 | equivalent to an end-of-file character. A common end-of-file character, <control>-D, is                                |
| 40727 | historically a <i>vi</i> command.                                                                                      |

40728 The text in the STANDARD OUTPUT section reflects the usage of the verb *display* in this section;  
 40729 some implementations of *vi* use standard output to write to the terminal, but  
 40730 IEEE Std. 1003.1-200x does not require that to be the case.

40731 Historically, implementations reverted to open mode if the terminal was incapable of  
 40732 supporting full visual mode. IEEE Std. 1003.1-200x requires this behavior. Historically, the open  
 40733 mode of *vi* behaved roughly equivalently to the visual mode, with the exception that only a  
 40734 single physical line from the edit buffer was kept current at any time. This line was normally  
 40735 displayed on the next-to-last line of a terminal with cursor addressing (and the last line  
 40736 performed its normal visual functions for line-oriented commands and messages). In addition,  
 40737 some few commands behaved differently in open mode than in visual mode.  
 40738 IEEE Std. 1003.1-200x requires conformance to historical practice.

40739 Historically, *ex* and *vi* implementations have expected text to proceed in the usual  
 40740 European/Latin order of left to right, top to bottom. There is no requirement in  
 40741 IEEE Std. 1003.1-200x that this be the case. The specification was deliberately written using  
 40742 words like “before”, “after”, “first”, and “last” in order to permit implementations to support  
 40743 the natural text order of the language.

40744 Historically, lines past the end of the edit buffer were marked with single tilde (‘~’) characters;  
 40745 that is, if the one-based display was 20 lines in length, and the last line of the file was on line one,  
 40746 then lines 2-20 would contain only a single ‘~’ character.

40747 Historically, the *vi* editor attempted to display only complete lines at the bottom of the screen (it  
 40748 did display partial lines at the top of the screen). If a line was too long to fit in its entirety at the  
 40749 bottom of the screen, the screen lines where the line would have been displayed were displayed  
 40750 as single ‘@’ characters, instead of displaying part of the line. IEEE Std. 1003.1-200x permits, but  
 40751 does not require, this behavior. Implementations are encouraged to attempt always to display a  
 40752 complete line at the bottom of the screen when doing scrolling or screen positioning by physical  
 40753 lines.

40754 Historically, lines marked with ‘@’ were also used to minimize output to dumb terminals over  
 40755 slow lines; that is, changes local to the cursor were updated, but changes to lines on the screen  
 40756 that were not close to the cursor were simply marked with an ‘@’ sign instead of being updated  
 40757 to match the current text. IEEE Std. 1003.1-200x permits, but does not require this feature  
 40758 because it is used ever less frequently as terminals become smarter and connections are faster.

#### 40759 Initialization in *ex* and *vi*

40760 Historically, *vi* always had a line in the edit buffer, even if the edit buffer was “empty”. For  
 40761 example:

- 40762 1. The *ex* command = executed from visual mode wrote “1” when the buffer was empty.
- 40763 2. Writes from visual mode of an empty edit buffer wrote files of a single character (a  
 40764 <newline> character), while writes from *ex* mode of an empty edit buffer wrote empty  
 40765 files.
- 40766 3. Put and read commands into an empty edit buffer left an empty line at the top of the edit  
 40767 buffer.

40768 For consistency, IEEE Std. 1003.1-200x does not permit any of these behaviors.

40769 Historically, *vi* did not always return the terminal to its original modes; for example, ICRNL was  
 40770 modified if it was not originally set. IEEE Std. 1003.1-200x does not permit this behavior.

## Command Descriptions in vi

Motion commands are among the most complicated aspects of *vi* to describe. With some exceptions, the text region and buffer type effect of a motion command on a *vi* command are described on a case-by-case basis. The descriptions of text regions in IEEE Std. 1003.1-200x are not intended to imply direction; that is, an inclusive region from line *n* to line *n*+5 is identical to a region from line *n*+5 to line *n*. This is of more than academic interest—movements to marks can be in either direction, and, if the **wrapscan** option is set, so can movements to search points. Historically, lines are always stored into buffers in text order; that is, from the start of the edit buffer to the end. IEEE Std. 1003.1-200x requires conformance to historical practice.

Historically, command counts were applied to any associated motion, and were multiplicative to any supplied motion count. For example, **2cw** is the same as **c2w**, and **2c3w** is the same as **c6w**. IEEE Std. 1003.1-200x requires this behavior. Historically, *vi* commands that used bigwords, words, paragraphs, and sentences as objects treated groups of empty lines, or lines that contained only <blank> characters, inconsistently. Some commands treated them as a single entity, while others treated each line separately. For example, the **w**, **W**, and **B** commands treated groups of empty lines as individual words; that is, the command would move the cursor to each new empty line. The **e** and **E** commands treated groups of empty lines as a single word; that is, the first use would move past the group of lines. The **b** command would just beep at the user, or if done from the start of the line as a motion command, fail in unexpected ways. If the lines contained only (or ended with) <blank> characters, the **w** and **W** commands would just beep at the user, the **E** and **e** commands would treat the group as a single word, and the **B** and **b** commands would treat the lines as individual words. For consistency and simplicity of specification, IEEE Std. 1003.1-200x requires that all *vi* commands treat groups of empty or <blank> character-filled lines as a single entity, and that movement through lines ending with <blank> characters be consistent with other movements.

Historically, *vi* documentation indicated that any number of double quotes were skipped after punctuation marks at sentence boundaries; however, implementations only skipped single quotes. IEEE Std. 1003.1-200x requires both to be skipped.

Historically, the first and last characters in the edit buffer were word boundaries. This historical practice is required by IEEE Std. 1003.1-200x.

Historically, *vi* attempted to update the minimum number of columns on the screen possible, which could lead to misleading information being displayed. IEEE Std. 1003.1-200x makes no requirements other than that the current character being entered is displayed correctly, leaving all other decisions in this area up to the implementation.

Historically, lines were arbitrarily folded between columns of any characters that required multiple column positions on the screen, with the exception of tabs, which terminated at the right-hand margin. IEEE Std. 1003.1-200x permits the former and requires the latter. Implementations that do not arbitrarily break lines between columns of characters that occupy multiple column positions should not permit the cursor to rest on a column that does not contain any part of a character.

The historical *vi* had a problem in that all movements were by physical lines, not by logical, or screen, lines. This is often the right thing to do; for example, single line movements, such as **j** or **k**, should work on physical lines. Commands like **dj**, or **j.**, where **.** is a change command, only make sense for physical lines. It is not, however, the right thing to do for screen motion or scrolling commands like <control>-D, <control>-F, and **H**. If the window is fairly small, using physical lines in these cases can result in completely random motion; for example, **1<control>-D** can result in a completely changed screen, without any overlap. This is clearly not what the user wanted. The problem is even worse in the case of the **H**, **L**, and **M** commands—as they position the cursor at the first non-<blank> character of the line, they may all refer to the same location in

40820 large lines, and will result in no movement at all.

40821 In addition, if the line is larger than the screen, using physical lines can make it impossible to  
40822 display parts of the line—there are not any commands that do not display the beginning of the  
40823 line in historical *vi*, and if both the beginning and end of the line cannot be on the screen at the  
40824 same time, the user suffers. Finally, the page and half-page scrolling commands historically  
40825 moved to the first non-<blank> character in the new line. If the line is approximately the same  
40826 size as the screen, this is inadequate because the cursor before and after a <control>-D command  
40827 will refer to the same location on the screen.

40828 Implementations of *ex* and *vi* exist that do not have these problems because the relevant  
40829 commands (<control>-B, <control>-D, <control>-F, <control>-U, <control>-Y, <control>-E, **H**, **L**,  
40830 and **M**) operate on logical screen lines, not physical edit buffer lines.

40831 IEEE Std. 1003.1-200x does not permit this behavior by default because the standard developers  
40832 believed that users would find it too confusing. However, historical practice has been relaxed.  
40833 For example, *ex* and *vi* historically attempted, albeit sometimes unsuccessfully, to never put part  
40834 of a line on the last lines of a screen; for example, if a line would not fit in its entirety, no part of  
40835 the line was displayed, and the screen lines corresponding to the line contained single '@'  
40836 characters. This behavior is permitted, but not required by IEEE Std. 1003.1-200x, so that it is  
40837 possible for implementations to support long lines in small screens more reasonably without  
40838 changing the commands to be logically (instead of physically) oriented. IEEE Std. 1003.1-200x  
40839 also permits implementations to refuse to edit any edit buffer containing a line that will not fit  
40840 on the screen in its entirety.

40841 The display area (for example, the value of the **window** edit option) has historically been  
40842 “grown”, or expanded, to display new text when local movements are done in displays where  
40843 the number of lines displayed is less than the maximum possible. Expansion has historically  
40844 been the first choice, when the target line is less than the maximum possible expansion value  
40845 away. Scrolling has historically been the next choice, done when the target line is less than half a  
40846 display away, and otherwise, the screen was redrawn. There were exceptions, however, in that  
40847 *ex* commands generally always caused the screen to be redrawn. IEEE Std. 1003.1-200x does not  
40848 specify a standard behavior because there may be external issues, such as connection speed, the  
40849 number of characters necessary to redraw as opposed to scroll, or terminal capabilities that  
40850 implementations will have to accommodate.

40851 The current line in IEEE Std. 1003.1-200x maps one-to-one to a physical line in the file. The  
40852 current column does not. There are two different column values that are described by  
40853 IEEE Std. 1003.1-200x. The first is the current column value as set by many of the *vi* commands.  
40854 This value is remembered for the lifetime of the editor. The second column value is the actual  
40855 position on the screen where the cursor rests. The two are not always the same. For example,  
40856 when the cursor is backed by a multi-column character, the actual cursor position on the screen  
40857 has historically been the last column of the character in command mode, and the first column of  
40858 the character in input mode.

40859 Commands that set the current line, but that do not set the current cursor value (for example, **j**  
40860 and **k**) attempt to get as close as possible to the remembered column position, so that the cursor  
40861 tends to restrict itself to a vertical column as the user moves around in the edit buffer.  
40862 IEEE Std. 1003.1-200x requires conformance to historical practice, requiring that the physical  
40863 location of the cursor on the screen be adjusted from the current column value as necessary to  
40864 support this historical behavior.

40865 Historically, only a single line (and for some terminals, a single line minus 1 column) of  
40866 characters could be entered by the user for the line oriented commands; that is, :, !, /, or ?.  
40867 IEEE Std. 1003.1-200x permits, but does not require, this limitation.

Historically, “soft” errors in *vi* caused the terminal to be alerted, but no error message was displayed. As a general rule, no error message was displayed for errors in command execution in *vi*, when the error resulted from the user attempting an invalid or impossible action, or when a searched-for object was not found. Examples of soft errors included **h** at the left margin, **<control>-B** or **[** at the beginning of the file, **2G** at the end of the file, and so on. In addition, errors such as **%**, **]**, **}**, **)**, **N**, **n**, **f**, **F**, **t**, and **T** failing to find the searched-for object were soft as well. Less consistently, **/** and **?** displayed an error message if the pattern was not found, **/**, **?**, **N**, and **n** displayed an error message if no previous regular expression had been specified, and **;** did not display an error message if no previous **f**, **F**, **t**, or **T** command had occurred. Also, behavior in this area might reasonably be based on a runtime evaluation of the speed of a network connection. Finally, some implementations have provided error messages for soft errors in order to assist naive users, based on the value of a verbose edit option. IEEE Std. 1003.1-200x does not list specific errors for which an error message shall be displayed. Implementations should conform to historical practice in the absence of any strong reason to diverge.

### Page Backwards

The **<control>-B** and **<control>-F** commands historically considered it an error to attempt to page past the beginning or end of the file, whereas the **<control>-D** and **<control>-U** commands simply moved to the beginning or end of the file. For consistency, IEEE Std. 1003.1-200x requires the latter behavior for all four commands. All four commands still consider it an error if the current line is at the beginning (**<control>-B**, **<control>-U**) or end (**<control>-F**, **<control>-D**) of the file. Historically, the **<control>-B** and **<control>-F** commands skip two lines in order to include overlapping lines when a single command is entered. This makes less sense in the presence of a *count*, as there will be, by definition, no overlapping lines. The actual calculation used by historical implementations of the *vi* editor for **<control>-B** was:

$((\text{current first line}) - \text{count } W \text{ (window edit option)}) + 2$

and for **<control>-F** was:

$((\text{current first line}) + \text{count } W \text{ (window edit option)}) - 2$

This calculation does not work well when intermixing commands with and without counts; for example, **3<control>-F** is not equivalent to entering the **<control>-F** command three times, and is not reversible by entering the **<control>-B** command three times. For consistency with other *vi* commands that take counts, IEEE Std. 1003.1-200x requires a different calculation.

### Scroll Forward

The 4BSD and System V implementations of *vi* differed on the initial value used by the **scroll** command. 4BSD used:

$((\text{window edit option}) + 1) / 2$

while System V used the value of the **scroll** edit option. The System V version is specified by IEEE Std. 1003.1-200x because the standard developers believed that it was more intuitive and permitted the user a method of setting the scroll value initially without also setting the number of lines that are displayed.

#### 40907 **Scroll Forward by Line**

40908 Historically, the <control>-E and <control>-Y commands considered it an error if the last and  
 40909 first lines, respectively, were already on the screen. IEEE Std. 1003.1-200x requires conformance  
 40910 to historical practice. Historically, the <control>-E and <control>-Y commands had no effect in  
 40911 open mode. For simplicity and consistency of specification, IEEE Std. 1003.1-200x requires that  
 40912 they behave as usual, albeit with a single line screen.

#### 40913 **Clear and Redisplay**

40914 The historical <control>-L command refreshed the screen exactly as it was supposed to be  
 40915 currently displayed, replacing any '@' characters for lines that had been deleted but not  
 40916 updated on the screen with refreshed '@' characters. The intent of the <control>-L command is  
 40917 to refresh when the screen has been accidentally overwritten; for example, by a **write** command  
 40918 from another user, or modem noise.

#### 40919 **Redraw Screen**

40920 The historical <control>-R command redisplayed only when necessary to update lines that had  
 40921 been deleted but not updated on the screen and that were flagged with '@' characters. There is  
 40922 no requirement that the screen be in any way refreshed if no lines of this form are currently  
 40923 displayed. IEEE Std. 1003.1-200x permits implementations to extend this command to refresh  
 40924 lines on the screen flagged with '@' characters because they are too long to be displayed in the  
 40925 current framework; however, the current line and column need not be modified.

#### 40926 **Search for tagstring**

40927 Historically, the first non-<blank> character at or after the cursor was the first character, and all  
 40928 subsequent characters that were word characters, up to the end of the line, were included. For  
 40929 example, with the cursor on the leading space or on the '#' character in the text "#bar@", the  
 40930 tag was "#bar". On the character 'b' it was "bar", and on the 'a', it was "ar".  
 40931 IEEE Std. 1003.1-200x requires this behavior.

#### 40932 **Replace Text with Results from Shell Command**

40933 Historically, the <, >, and ! commands considered most cursor motions other than line-oriented  
 40934 motions an error; for example, the command >/foo<CR> succeeded, while the command >|  
 40935 failed, even though the text region described by the two commands might be identical. For  
 40936 consistency, all three commands only consider entire lines and not partial lines, and the region is  
 40937 defined as any line that contains a character that was specified by the motion.

#### 40938 **Move to Matching Character**

40939 Other matching characters have been left implementation-dependent in order to allow  
 40940 extensions such as matching '<' and '>' for searching HTML, or **#ifdef**, **#else**, and **#endif** for  
 40941 searching C source.

## Repeat Substitution

IEEE Std. 1003.1-200x requires that any **c** and **g** flags specified to the previous substitute command be ignored; however, the **r** flag may still apply, if supported by the implementation.

## Return to Previous (Context or Section)

The **[[**, **]]**, **(**, **)**, **{**, and **}** commands are all affected by “section boundaries”, but in some historical implementations not all of the commands recognize the same section boundaries. This is a bug, not a feature, and a unique section-boundary algorithm was not described for each command. One special case that is preserved is that the sentence command moves to the end of the last line of the edit buffer while the other commands go to the beginning, in order to preserve the traditional character cut semantics of the sentence command. Historically, *vi* section boundaries at the beginning and end of the edit buffer were the first non-<blank> character on the first and last lines of the edit buffer if one exists; otherwise, the last character of the first and last lines of the edit buffer if one exists. To increase consistency with other section locations, this has been simplified by IEEE Std. 1003.1-200x to the first character of the first and last lines of the edit buffer, or the first and the last lines of the edit buffer if they are empty.

Sentence boundaries were problematic in the historical *vi*. They were not only the boundaries as defined for the section and paragraph commands, but they were the first non-<blank> character that occurred after those boundaries, as well. Historically, the *vi* section commands were documented as taking an optional window size as a *count* preceding the command. This was not implemented in historical versions, so IEEE Std. 1003.1-200x requires that the *count* repeat the command, for consistency with other *vi* commands.

## Repeat

Historically, mapped commands other than text input commands could not be repeated using the **period** command. IEEE Std. 1003.1-200x requires conformance to historical practice.

The restrictions on the interpretation of special characters (for example, <control>-H) in the repetition of text input mode commands is intended to match historical practice. For example, given the input sequence:

```
iab<control>-H<control>-H<control>-Hdef<escape>
```

the user should be informed of an error when the sequence is first entered, but not during a command repetition. The character <control>-T is specifically exempted from this restriction. Historical implementations of *vi* ignored <control>-T characters that were input in the original command during command repetition. IEEE Std. 1003.1-200x prohibits this behavior.

## Find Regular Expression

Historically, commands did not affect the line searched to or from if the motion command was a search (**/**, **?**, **N**, **n**) and the final position was the start/end of the line. There were some special cases and *vi* was not consistent. IEEE Std. 1003.1-200x does not permit this behavior, for consistency. Historical implementations permitted, but were unable to handle searches as motion commands that wrapped (that is, due to the edit option **wrapsan**) to the original location. IEEE Std. 1003.1-200x requires that this behavior be treated as an error.

Historically, the syntax **"/RE/0"** was used to force the command to cut text in line mode. IEEE Std. 1003.1-200x requires conformance to historical practice.

Historically, in open mode, a **z** specified to a search command redisplayed the current line instead of displaying the current screen with the current line highlighted. For consistency and simplicity of specification, IEEE Std. 1003.1-200x does not permit this behavior.



Historically, trailing **z** commands were permitted and ignored if entered as part of a search used as a motion command. For consistency and simplicity of specification, IEEE Std. 1003.1-200x does not permit this behavior.

#### Execute an **ex** Command

Historically, **vi** implementations restricted the commands that could be entered on the colon command line (for example, **append** and **change**), and some other commands were known to cause them to fail catastrophically. For consistency, IEEE Std. 1003.1-200x does not permit these restrictions. When executing an **ex** command by entering **:**, it is not possible to enter a <newline> character as part of the command because it is considered the end of the command. A different approach is to enter **ex** command mode by using the **vi Q** command (and later resuming visual mode with the **ex vi** command). In **ex** command mode, the single-line limitation does not exist. So, for example, the following is valid:

```
Q
s/break here/break\
here/
vi
```

IEEE Std. 1003.1-200x requires that, if the **ex** command overwrites any part of the screen that would be erased by a refresh, **vi** pauses for a character from the user. Historically, this character could be any character; for example, a character input by the user before the message appeared, or even a mapped character. This is probably a bug, but implementations that have tried to be more rigorous by requiring that the user enter a specific character, or that the user enter a character after the message was displayed, have been forced by user indignation back into historical behavior. IEEE Std. 1003.1-200x requires conformance to historical practice.

#### Shift Left (Right)

Refer to the Rationale for the **!** and **/** commands. Historically, the **<** and **>** commands sometimes moved the cursor to the first non-<blank> character (for example if the command was repeated or with **\_** as the motion command), and sometimes left it unchanged. IEEE Std. 1003.1-200x does not permit this inconsistency, requiring instead that the cursor always move to the first non-<blank> character. Historically, the **<** and **>** commands did not support buffer arguments, although some implementations allow the specification of an optional buffer. This behavior is neither required nor disallowed by IEEE Std. 1003.1-200x.

#### Execute

Historically, buffers could execute other buffers, and loops, infinite and otherwise, were possible. IEEE Std. 1003.1-200x requires conformance to historical practice. The *\*buffer* syntax of **ex** is not required in **vi**, because it is not historical practice and has been used in some **vi** implementations to support additional scripting languages.

Historically, **vi** only supported the **"@@"** syntax for repeating the last buffer execution. IEEE Std. 1003.1-200x requires that **vi** support the additional **ex** syntax **"@\*"** as well, for consistency.

**41025 Reverse Case**

41026 Historically, the `~` command ignored any associated *count*, and acted only on the characters in  
41027 the current line. For consistency with other *vi* commands, IEEE Std. 1003.1-200x requires that an  
41028 associated *count* act on the next *count* characters, and that the command move to subsequent  
41029 lines if warranted by *count*, to make it possible to modify large pieces of text in a reasonably  
41030 efficient manner. There exist *vi* implementations that optionally require an associated motion  
41031 command for the `~` command. Implementations supporting this functionality are encouraged to  
41032 base it on the **tildedop** edit option and handle the text regions and cursor positioning identically  
41033 to the **yank** command.

**41034 Append**

41035 Historically, *counts* specified to the **A**, **a**, **I**, and **i** commands repeated the input of the first line  
41036 *count* times, and did not repeat the subsequent lines of the input text. IEEE Std. 1003.1-200x  
41037 requires that the entire text input be repeated *count* times.

**41038 Move Backward to Preceding Word**

41039 Historically, *vi* became confused if word commands were used as motion commands in empty  
41040 files. IEEE Std. 1003.1-200x requires that this be an error. Historical implementations of *vi* had a  
41041 large number of bugs in the word movement commands, and they varied greatly in behavior in  
41042 the presence of empty lines, “words” made up of a single character, and lines containing only  
41043 <blank> characters. For consistency and simplicity of specification, IEEE Std. 1003.1-200x does  
41044 not permit this behavior.

**41045 Change to End-of-Line**

41046 Some historical implementations of the **C** command did not behave as described by  
41047 IEEE Std. 1003.1-200x when the **\$** key was remapped because they were implemented by  
41048 pushing the **\$** key onto the input queue and reprocessing it. IEEE Std. 1003.1-200x does not  
41049 permit this behavior. Historically, the **C**, **S**, and **s** commands did not copy replaced text into the  
41050 numeric buffers. For consistency and simplicity of specification, IEEE Std. 1003.1-200x requires  
41051 that they behave like their respective **c** commands in all respects.

**41052 Delete**

41053 Historically, lines in open mode that were deleted were scrolled up, and an @ glyph written over  
41054 the beginning of the line. In the case of terminals that are incapable of the necessary cursor  
41055 motions, the editor erased the deleted line from the screen. IEEE Std. 1003.1-200x requires  
41056 conformance to historical practice; that is, if the terminal cannot display the '@' character, the  
41057 line cannot remain on the screen.

**41058 Delete to End-of-Line**

41059 Some historical implementations of the **D** command did not behave as described by  
41060 IEEE Std. 1003.1-200x when the **\$** key was remapped because they were implemented by  
41061 pushing the **\$** key onto the input queue and reprocessing it. IEEE Std. 1003.1-200x does not  
41062 permit this behavior.

41063 **Join**

41064 An historical oddity of *vi* is that the commands **J**, **1J**, and **2J** are all equivalent.  
 41065 IEEE Std. 1003.1-200x requires conformance to historical practice. The *vi* **J** command is specified  
 41066 in terms of the *ex* **join** command with an *ex* command *count* value. The address correction for a  
 41067 *count* that is past the end of the edit buffer is necessary for historical compatibility for both *ex*  
 41068 and *vi*.

41069 **Mark Position**

41070 Historical practice is that only lowercase letters, plus `'''` and `'''`, could be used to mark a  
 41071 cursor position. IEEE Std. 1003.1-200x requires conformance to historical practice, but  
 41072 encourages implementations to support other characters as marks as well.

41073 **Repeat Regular Expression Find (Forward and Reverse)**

41074 Historically, the **N** and **n** commands could not be used as motion components for the **c**  
 41075 command. With the exception of the **cN** command, which worked if the search crossed a line  
 41076 boundary, the text region would be discarded, and the user would not be in text input mode. For  
 41077 consistency and simplicity of specification, IEEE Std. 1003.1-200x does not permit this behavior.

41078 **Insert Empty Line (Below and Above)**

41079 Historically, counts to the **O** and **o** commands were used as the number of physical lines to  
 41080 open, if the terminal was dumb and the **slowopen** option was not set. This was intended to  
 41081 minimize traffic over slow connections and repainting for dumb terminals. IEEE Std. 1003.1-200x  
 41082 does not permit this behavior, requiring that a *count* to the open command behave as for other  
 41083 text input commands. This change to historical practice was made for consistency, and because a  
 41084 superset of the functionality is provided by the **slowopen** edit option.

41085 **Put from Buffer (Following and Before)**

41086 Historically, *counts* to the **p** and **P** commands were ignored if the buffer was a line mode buffer,  
 41087 but were (mostly) implemented as described in IEEE Std. 1003.1-200x if the buffer was a  
 41088 character mode buffer. Because implementations exist that do not have this limitation, and  
 41089 because pasting lines multiple times is generally useful, IEEE Std. 1003.1-200x requires that *count*  
 41090 be supported for all **p** and **P** commands.

41091 Historical implementations of *vi* were widely known to have major problems in the **p** and **P**  
 41092 commands, particularly when unusual regions of text were copied into the edit buffer. The  
 41093 standard developers viewed these as bugs, and they are not permitted for consistency and  
 41094 simplicity of specification.

41095 Historically, a **P** or **p** command (or an *ex* **put** command executed from open or visual mode)  
 41096 executed in an empty file, left an empty line as the first line of the file. For consistency and  
 41097 simplicity of specification,  $\zeta$ A does not permit this behavior.

41098 **Replace Character**

41099 Historically, the **r** command did not correctly handle the *erase* and *word erase* characters as  
 41100 arguments, nor did it handle an associated *count* greater than 1 with a <carriage-return>  
 41101 character argument, for which it replaced *count* characters with a single <newline> character.  
 41102 IEEE Std. 1003.1-200x does not permit these inconsistencies.

41103 Historically, the **r** command permitted the <control>-V escaping of entered characters, such as  
 41104 <ESC> and the <carriage-return> character; however, it required two leading <control>-V  
 41105 characters instead of one. IEEE Std. 1003.1-200x requires that this be changed for consistency

41106 with the other text input commands of *vi*.

41107 Historically, it is an error to enter the **r** command if there are less than *count* characters at or after  
 41108 the cursor in the line. While a reasonable and unambiguous extension would be to permit the **r**  
 41109 command on empty lines, it would require that too large a *count* be adjusted to match the  
 41110 number of characters at or after the cursor for consistency, which is sufficiently different from  
 41111 historical practice to be avoided. IEEE Std. 1003.1-200x requires conformance to historical  
 41112 practice.

#### 41113 **Replace Characters**

41114 Historically, if there were **autoindent** characters in the line on which the **R** command was run,  
 41115 and **autoindent** was set, the first <newline> character would be properly indented and no  
 41116 characters would be replaced by the <newline> character. Each additional <newline> character  
 41117 would replace *n* characters, where *n* was the number of characters that were needed to indent  
 41118 the rest of the line to the proper indentation level. This behavior is a bug and is not permitted by  
 41119 IEEE Std. 1003.1-200x.

#### 41120 **Undo**

41121 Historical practice for cursor positioning after undoing commands was mixed. In most cases,  
 41122 when undoing commands that affected a single line, the cursor was moved to the start of added  
 41123 or changed text, or immediately after deleted text. However, if the user had moved from the line  
 41124 being changed, the column was either set to the first non-<blank> character, returned to the  
 41125 origin of the command, or remained unchanged. When undoing commands that affected  
 41126 multiple lines or entire lines, the cursor was moved to the first character in the first line restored.  
 41127 As an example of how inconsistent this was, a search, followed by an **o** text input command,  
 41128 followed by an **undo** would return the cursor to the location where the **o** command was entered,  
 41129 but a **cw** command followed by an **o** command followed by an **undo** would return the cursor to  
 41130 the first non-<blank> character of the line. IEEE Std. 1003.1-200x requires the most useful of  
 41131 these behaviors, and discards the least useful, in the interest of consistency and simplicity of  
 41132 specification.

#### 41133 **Yank**

41134 Historically, the **yank** command did not move to the end of the motion if the motion was in the  
 41135 forward direction. It moved to the end of the motion if the motion was in the backward  
 41136 direction, except for the **\_** command, or for the **G** and **'** commands when the end of the motion  
 41137 was on the current line. This was further complicated by the fact that for a number of motion  
 41138 commands, the **yank** command moved the cursor but did not update the screen; for example, a  
 41139 subsequent command would move the cursor from the end of the motion, even though the  
 41140 cursor on the screen had not reflected the cursor movement for the **yank** command.  
 41141 IEEE Std. 1003.1-200x requires that all **yank** commands associated with backward motions move  
 41142 the cursor to the end of the motion for consistency, and specifically, to make **'** commands as  
 41143 motions consistent with search patterns as motions.

41144 **Yank Current Line**

41145 Some historical implementations of the **Y** command did not behave as described by  
 41146 IEEE Std. 1003.1-200x when the `'_'` key was remapped because they were implemented by  
 41147 pushing the `'_'` key onto the input queue and reprocessing it. IEEE Std. 1003.1-200x does not  
 41148 permit this behavior.

41149 **Redraw Window**

41150 Historically, the **z** command always redrew the screen. This is permitted but not required by  
 41151 IEEE Std. 1003.1-200x, because of the frequent use of the **z** command in macros such as **map n**  
 41152 **nz**, for screen positioning, instead of its use to change the screen size. The standard developers  
 41153 believed that expanding or scrolling the screen offered a better interface for users. The ability to  
 41154 redraw the screen is preserved if the optional new window size is specified, and in the  
 41155 `<control>-L` and `<control>-R` commands.

41156 The semantics of **z^** are confusing at best. Historical practice is that the screen before the screen  
 41157 that ended with the specified line is displayed. IEEE Std. 1003.1-200x requires conformance to  
 41158 historical practice.

41159 Historically, the **z** command would not display a partial line at the top or bottom of the screen. If  
 41160 the partial line would normally have been displayed at the bottom of the screen, the command  
 41161 worked, but the partial line was replaced with `'@'` characters. If the partial line would normally  
 41162 have been displayed at the top of the screen, the command would fail. For consistency and  
 41163 simplicity of specification, IEEE Std. 1003.1-200x does not permit this behavior.

41164 Historically, the **z** command with a line specification of 1 ignored the command. For consistency  
 41165 and simplicity of specification, IEEE Std. 1003.1-200x does not permit this behavior.

41166 Historically, the **z** command did not set the cursor column to the first non-`<blank>` character for  
 41167 the character if the first screen was to be displayed, and was already displayed. For consistency  
 41168 and simplicity of specification, IEEE Std. 1003.1-200x does not permit this behavior.

41169 **Input Mode Commands in vi**

41170 Historical implementations of **vi** did not permit the the user to erase more than a single line of  
 41171 input, or to use normal erase characters such as *line erase*, *worderase*, and *erase* to erase  
 41172 **autoindent** characters. As there exist implementations of **vi** that do not have these limitations,  
 41173 both behaviors are permitted, but only historical practice is required. In the case of these  
 41174 extensions, **vi** is required to pause at the **autoindent** and previous line boundaries.

41175 Historical implementations of **vi** updated only the portion of the screen where the current cursor  
 41176 character was displayed. For example, consider the **vi** input keystrokes:

41177 `iabcd<escape>0C<tab>`

41178 Historically, the `<tab>` character would overwrite the characters `"abcd"` when it was displayed.  
 41179 Other implementations replace only the `'a'` character with the `<tab>` character, and then push  
 41180 the rest of the characters ahead of the cursor. Both implementations have problems. The  
 41181 historical implementation is probably visually nicer for the above example; however, for the  
 41182 keystrokes:

41183 `iabcd<ESC>0R<tab><ESC>`

41184 the historical implementation results in the string `"bcd"` disappearing and then magically  
 41185 reappearing when the `<ESC>` character is entered. IEEE Std. 1003.1-200x requires the former  
 41186 behavior when overwriting erase-columns; that is, overwriting characters that are no longer  
 41187 logically part of the edit buffer, and the latter behavior otherwise.

41188 Historical implementations of *vi* discarded the <control>-D and <control>-T characters when  
 41189 they were entered at places where their command functionality was not appropriate.  
 41190 IEEE Std. 1003.1-200x requires that the <control>-T functionality always be available, and that  
 41191 <control>-D be treated as any other key when not operating on **autoindent** characters.

## 41192 NUL

41193 Some historical implementations of *vi* limited the number of characters entered using the NUL  
 41194 input character to 256 bytes. IEEE Std. 1003.1-200x permits this limitation; however,  
 41195 implementations are encouraged to remove this limit.

## 41196 <control>-D

41197 See also Rationale for the input mode command <newline>. The hidden assumptions in the  
 41198 <control>-D command (and in the *vi* **autoindent** specification in general) is that <space>  
 41199 characters take up a single column on the screen and that <tab> characters are comprised of an  
 41200 integral number of <space> characters.

## 41201 <newline>

41202 Implementations are permitted to rewrite **autoindent** characters in the line when <newline>,  
 41203 <carriage-return>, <control>-D, and <control>-T are entered, or when the **shift** commands are  
 41204 used, because historical implementations have both done so and found it necessary to do so. For  
 41205 example, a <control>-D when the cursor is preceded by a single <tab> character, with **tabstop**  
 41206 set to 8, and **shiftwidth** set to 3, will result in the <tab> character being replaced by several  
 41207 <space> characters.

## 41208 <control>-T

41209 See also the Rationale for the input mode command <newline>. Historically, <control>-T only  
 41210 worked if no non-<blank> characters had yet been input in the current input line. In addition,  
 41211 the characters inserted by <control>-T were treated as **autoindent** characters, and could not be  
 41212 erased using normal user erase characters. Because implementations exist that do not have  
 41213 these limitations, and as moving to a column boundary is generally useful, IEEE Std. 1003.1-200x  
 41214 requires that both limitations be removed.

## 41215 <control>-V

41216 Historically, *vi* used ^V, regardless of the value of the literal-next character of the terminal.  
 41217 IEEE Std. 1003.1-200x requires conformance to historical practice.

41218 The uses described for <control>-V can also be accomplished with <control>-Q, which is useful  
 41219 on terminals that use <control>-V for the down-arrow function. However, most historical  
 41220 implementations use <control>-Q for the *termios* START character, so the editor will generally  
 41221 not receive the <control>-Q unless **stty ixon** mode is set to off. (In addition, some historical  
 41222 implementations of *vi* explicitly set **ixon** mode to on, so it was difficult for the user to set it to  
 41223 off.) Any of the command characters described in IEEE Std. 1003.1-200x can be made ineffective  
 41224 by their selection as *termios* control characters, using the *stty* utility or other methods described  
 41225 in the System Interfaces volume of IEEE Std. 1003.1-200x.

|       |                                                                                                       |  |
|-------|-------------------------------------------------------------------------------------------------------|--|
| 41226 | <ESC>                                                                                                 |  |
| 41227 | Historically, SIGINT alerted the terminal when used to end input mode. This behavior is               |  |
| 41228 | permitted, but not required, by IEEE Std. 1003.1-200x.                                                |  |
| 41229 | <b>FUTURE DIRECTIONS</b>                                                                              |  |
| 41230 | None.                                                                                                 |  |
| 41231 | <b>SEE ALSO</b>                                                                                       |  |
| 41232 | <i>ex</i> , <i>stty</i>                                                                               |  |
| 41233 | <b>CHANGE HISTORY</b>                                                                                 |  |
| 41234 | First released in Issue 2.                                                                            |  |
| 41235 | <b>Issue 4</b>                                                                                        |  |
| 41236 | Aligned with the ISO/IEC 9945-2: 1993 standard.                                                       |  |
| 41237 | <b>Issue 5</b>                                                                                        |  |
| 41238 | FUTURE DIRECTIONS section added.                                                                      |  |
| 41239 | <b>Issue 6</b>                                                                                        |  |
| 41240 | This utility is now marked as part of the User Portability Utilities option.                          |  |
| 41241 | The APPLICATION USAGE section is added.                                                               |  |
| 41242 | The obsolescent SYNOPSIS is removed.                                                                  |  |
| 41243 | The following new requirements on POSIX implementations derive from alignment with the                |  |
| 41244 | Single UNIX Specification:                                                                            |  |
| 41245 | <ul style="list-style-type: none"> <li>• Lisp mode is added.</li> </ul>                               |  |
| 41246 | <ul style="list-style-type: none"> <li>• The <b>reindent</b> command description is added.</li> </ul> |  |
| 41247 | The <i>vi</i> utility has been extensively rewritten for alignment with the IEEE P1003.2b draft       |  |
| 41248 | standard.                                                                                             |  |

41249 **NAME**

41250           wait — await process completion

41251 **SYNOPSIS**41252           wait [*pid*...]41253 **DESCRIPTION**

41254           When an asynchronous list (see Section 2.9.3.1 on page 74) is started by the shell, the process ID  
 41255           of the last command in each element of the asynchronous list shall become known in the current  
 41256           shell execution environment; see Section 2.12 on page 90.

41257           If the *wait* utility is invoked with no operands, it shall wait until all process IDs known to the  
 41258           invoking shell have terminated and exit with a zero exit status.

41259           If one or more *pid* operands are specified that represent known process IDs, the *wait* utility shall  
 41260           wait until all of them have terminated. If one or more *pid* operands are specified that represent  
 41261           unknown process IDs, *wait* shall treat them as if they were known process IDs that exited with  
 41262           exit status 127. The exit status returned by the *wait* utility shall be the exit status of the process  
 41263           requested by the last *pid* operand.

41264           The known process IDs are applicable only for invocations of *wait* in the current shell execution  
 41265           environment.

41266 **OPTIONS**

41267           None.

41268 **OPERANDS**

41269           The following operand shall be supported:

41270           *pid*           One of the following:

41271                           1. The unsigned decimal integer process ID of a command, for which the utility  
 41272                           is to wait for the termination.

41273                           2. A job control job ID (see the System Interface Definitions volume of  
 41274                           IEEE Std. 1003.1-200x, Section 3.207, Job Control Job ID) that identifies a  
 41275                           background process group to be waited for. The job control job ID notation is  
 41276                           applicable only for invocations of *wait* in the current shell execution  
 41277                           environment; see Section 2.12 on page 90. The exit status of *wait* shall be  
 41278                           determined by the last command in the pipeline.

41279                           **Note:**       The job control job ID type of *pid* is only available on systems  
 41280                           supporting the User Portability Utilities option.

41281 **STDIN**

41282           Not used.

41283 **INPUT FILES**

41284           None.

41285 **ENVIRONMENT VARIABLES**41286           The following environment variables shall affect the execution of *wait*:

41287           *LANG*           Provide a default value for the internationalization variables that are unset or null.  
 41288                           If *LANG* is unset or null, the corresponding value from the implementation-  
 41289                           dependent default locale shall be used. If any of the internationalization variables  
 41290                           contains an invalid setting, the utility shall behave as if none of the variables had  
 41291                           been defined.



41292 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
 41293 internationalization variables.

41294 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
 41295 characters (for example, single-byte as opposed to multi-byte characters in  
 41296 arguments).

41297 **LC\_MESSAGES**  
 41298 Determine the locale that should be used to affect the format and contents of  
 41299 diagnostic messages written to standard error.

41300 XSI **NLSPATH** Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

41301 **ASYNCHRONOUS EVENTS**  
 41302 Default.

41303 **STDOUT**  
 41304 Not used.

41305 **STDERR**  
 41306 Used only for diagnostic messages.

41307 **OUTPUT FILES**  
 41308 None.

41309 **EXTENDED DESCRIPTION**  
 41310 None.

41311 **EXIT STATUS**  
 41312 If one or more operands were specified, all of them have terminated or were not known by the  
 41313 invoking shell, and the status of the last operand specified is known, then the exit status of *wait*  
 41314 shall be the exit status information of the command indicated by the last operand specified. If  
 41315 the process terminated abnormally due to the receipt of a signal, the exit status shall be greater  
 41316 than 128 and shall be distinct from the exit status generated by other signals, but the exact value  
 41317 is unspecified. (See the *kill -l* option.) Otherwise, the *wait* utility shall exit with one of the  
 41318 following values:

41319 0 The *wait* utility was invoked with no operands and all process IDs known by the  
 41320 invoking shell have terminated.

41321 1-126 The *wait* utility detected an error.

41322 127 The command identified by the last *pid* operand specified is unknown.

41323 **CONSEQUENCES OF ERRORS**  
 41324 Default.

41325 **APPLICATION USAGE**  
 41326 On most implementations, *wait* is a shell built-in. If it is called in a subshell or separate utility  
 41327 execution environment, such as one of the following:

41328 (wait)  
 41329 nohup wait ...  
 41330 find . -exec wait ... \;

41331 it returns immediately because there are no known process IDs to wait for in those  
 41332 environments.

41333 Historical implementations of interactive shells have discarded the exit status of terminated  
 41334 background processes before each shell prompt. Therefore, the status of background processes  
 41335 was usually lost unless it terminated while *wait* was waiting for it. This could be a serious

problem when a job that was expected to run for a long time actually terminated quickly with a syntax or initialization error because the exit status returned was usually zero if the requested process ID was not found. This volume of IEEE Std. 1003.1-200x requires the implementation to keep the status of terminated jobs available until the status is requested, so that scripts like:

```

41340 j1&
41341 p1=$!
41342 j2&
41343 wait $p1
41344 echo Job 1 exited with status $?
41345 wait $!
41346 echo Job 2 exited with status $?

```

works without losing status on any of the jobs. The shell is allowed to discard the status of any process that it determines the application cannot get the process ID from the shell. It is also required to remember only {CHILD\_MAX} number of processes in this way. Since the only way to get the process ID from the shell is by using the '!' shell parameter, the shell is allowed to discard the status of an asynchronous list if "\$!" was not referenced before another asynchronous list was started. (This means that the shell only has to keep the status of the last asynchronous list started if the application did not reference "\$!". If the implementation of the shell is smart enough to determine that a reference to "\$!" was not saved anywhere that the application can retrieve it later, it can use this information to trim the list of saved information. Note also that a successful call to *wait* with no operands discards the exit status of all asynchronous lists.)

If the exit status of *wait* is greater than 128, there is no way for the application to know if the waited-for process exited with that value or was killed by a signal. Since most utilities exit with small values, there is seldom any ambiguity. Even in the ambiguous cases, most applications just need to know that the asynchronous job failed; it does not matter whether it detected an error and failed or was killed and did not complete its job normally.

#### 41363 EXAMPLES

Although the exact value used when a process is terminated by a signal is unspecified, if it is known that a signal terminated a process, a script can still reliably figure out which signal using *kill* as shown by the following script:

```

41367 sleep 1000&
41368 pid=$!
41369 kill -kill $pid
41370 wait $pid
41371 echo $pid was terminated by a SIG$(kill -l $?) signal.

```

If the following sequence of commands is run in less than 31 seconds:

```

41373 sleep 257 | sleep 31 &
41374 jobs -l %%

```

either of the following commands returns the exit status of the second *sleep* in the pipeline:

```

41376 wait <pid of sleep 31>
41377 wait %%

```

#### 41378 RATIONALE

The description of *wait* does not refer to the *waitpid()* function from the System Interfaces volume of IEEE Std. 1003.1-200x because that would needlessly overspecify this interface. However, the wording means that *wait* is required to wait for an explicit process when it is given an argument so that the status information of other processes is not consumed. Historical

41383 implementations use The *wait()* function defined in the System Interfaces volume of  
41384 IEEE Std. 1003.1-200x until *wait()* returns the requested process ID or finds that the requested  
41385 process does not exist. Because this means that a shell script could not reliably get the status of  
41386 all background children if a second background job was ever started before the first job finished,  
41387 it is recommended that the *wait* utility use a method such as the functionality provided by the  
41388 *waitpid()* function.

41389 The ability to wait for multiple *pid* operands was adopted from the KornShell.

41390 This new functionality was added because it is needed to determine the exit status of any  
41391 asynchronous list accurately. The only compatibility problem that this change creates is for a  
41392 script like

```
41393 while sleep 60 do
41394 job& echo Job started $(date) as $! done
```

41395 which causes the shell to monitor all of the jobs started until the script terminates or runs out of  
41396 memory. This would not be a problem if the loop did not reference "\$!" or if the script would  
41397 occasionally *wait* for jobs it started.

#### 41398 **FUTURE DIRECTIONS**

41399 None.

#### 41400 **SEE ALSO**

41401 *sh*, the System Interfaces volume of IEEE Std. 1003.1-200x, *waitpid()*

#### 41402 **CHANGE HISTORY**

41403 First released in Issue 2.

#### 41404 **Issue 4**

41405 Aligned with the ISO/IEC 9945-2: 1993 standard.

41406 **NAME**

41407           wc — word, line, and byte or character count

41408 **SYNOPSIS**41409           wc [-c|-m][-lw][*file...*]41410 **DESCRIPTION**41411           The *wc* utility shall read one or more input files and, by default, write the number of <newline>  
41412 characters, words, and bytes contained in each input file to the standard output.41413           The utility also shall write a total count for all named files, if more than one input file is  
41414 specified.41415           The *wc* utility shall consider a *word* to be a non-zero-length string of characters delimited by  
41416 white space.41417 **OPTIONS**41418           The *wc* utility shall conform to the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
41419 Section 12.2, Utility Syntax Guidelines.

41420           The following options shall be supported:

41421           **-c**           Write to the standard output the number of bytes in each input file.41422           **-l**           Write to the standard output the number of <newline> characters in each input  
41423 file.41424           **-m**           Write to the standard output the number of characters in each input file.41425           **-w**           Write to the standard output the number of words in each input file.41426           When any option is specified, *wc* shall report only the information requested by the specified  
41427 options.41428 **OPERANDS**

41429           The following operand shall be supported:

41430           *file*           A path name of an input file. If no *file* operands are specified, the standard input  
41431 shall be used.41432 **STDIN**41433           The standard input shall be used only if no *file* operands are specified. See the INPUT FILES  
41434 section.41435 **INPUT FILES**

41436           The input files may be of any type.

41437 **ENVIRONMENT VARIABLES**41438           The following environment variables shall affect the execution of *wc*:41439           **LANG**           Provide a default value for the internationalization variables that are unset or null.  
41440           If *LANG* is unset or null, the corresponding value from the implementation-  
41441 dependent default locale shall be used. If any of the internationalization variables  
41442 contains an invalid setting, the utility shall behave as if none of the variables had  
41443 been defined.41444           **LC\_ALL**          If set to a non-empty string value, override the values of all the other  
41445 internationalization variables.41446           **LC\_CTYPE**       Determine the locale for the interpretation of sequences of bytes of text data as  
41447 characters (for example, single-byte as opposed to multi-byte characters in  
41448 arguments and input files) and which characters are defined as white space

- 41449 characters.
- 41450 **LC\_MESSAGES**
- 41451 Determine the locale that should be used to affect the format and contents of
- 41452 diagnostic messages written to standard error and informative messages written to
- 41453 standard output.
- 41454 XSI **NLSPATH** Determine the location of message catalogs for the processing of **LC\_MESSAGES**.
- 41455 **ASYNCHRONOUS EVENTS**
- 41456 Default.
- 41457 **STDOUT**
- 41458 By default, the standard output shall contain an entry for each input file of the form:
- 41459 "%d %d %d %s\n", <newlines>, <words>, <bytes>, <file>
- 41460 If the **-m** option is specified, the number of characters shall replace the <bytes> field in this
- 41461 format.
- 41462 If any options are specified and the **-l** option is not specified, the number of <newline>
- 41463 characters shall not be written.
- 41464 If any options are specified and the **-w** option is not specified, the number of words shall not be
- 41465 written.
- 41466 If any options are specified and neither **-c** nor **-m** is specified, the number of bytes or characters
- 41467 shall not be written.
- 41468 If no input *file* operands are specified, no name shall be written and no <blank> characters
- 41469 preceding the path name shall be written.
- 41470 If more than one input *file* operand is specified, an additional line shall be written, of the same
- 41471 format as the other lines, except that the word **total** (in the POSIX locale) shall be written instead
- 41472 of a path name and the total of each column shall be written as appropriate. Such an additional
- 41473 line, if any, is written at the end of the output.
- 41474 **STDERR**
- 41475 Used only for diagnostic messages.
- 41476 **OUTPUT FILES**
- 41477 None.
- 41478 **EXTENDED DESCRIPTION**
- 41479 None.
- 41480 **EXIT STATUS**
- 41481 The following exit values shall be returned:
- 41482 0 Successful completion.
- 41483 >0 An error occurred.
- 41484 **CONSEQUENCES OF ERRORS**
- 41485 Default.

**41486 APPLICATION USAGE**

41487       The **-m** option is not a switch, but an option at the same level as **-c**. Thus, to produce the full  
41488       default output with character counts instead of bytes, the command required is:

41489       wc -mlw

**41490 EXAMPLES**

41491       None.

**41492 RATIONALE**

41493       The output file format pseudo-*printf()* string differs from the the System V version of *wc*:

41494       "%7d%7d%7d %s\n"

41495       which produces possibly ambiguous and unparsable results for very large files, as it assumes no  
41496       number shall exceed six digits.

41497       Some historical implementations use only <space>, <tab>, and <newline> as word separators.  
41498       The equivalent of the ISO C standard *isspace()* function is more appropriate.

41499       The **-c** option stands for “character” count, even though it counts bytes. This stems from the  
41500       sometimes erroneous historical view that bytes and characters are the same size. Due to  
41501       international requirements, the **-m** option (reminiscent of “multi-byte”) was added to obtain  
41502       actual character counts.

41503       Early proposals only specified the results when input files were text files. The current  
41504       specification more closely matches historical practice. (Bytes, words, and <newline>s are  
41505       counted separately and the results are written when an end-of-file is detected.)

41506       Historical implementations of the *wc* utility only accepted one argument to specify the options  
41507       **-c**, **-l**, and **-w**. Some of them also had multiple occurrences of an option cause the  
41508       corresponding count to be written multiple times and had the order of specification of the  
41509       options affect the order of the fields on output, but did not document either of these. Because  
41510       common usage either specifies no options or only one option, and because none of this was  
41511       documented, the changes required by this volume of IEEE Std. 1003.1-200x should not break  
41512       many historical applications (and do not break any historical portable applications).

**41513 FUTURE DIRECTIONS**

41514       None.

**41515 SEE ALSO**

41516       *cksum*

**41517 CHANGE HISTORY**

41518       First released in Issue 2.

**41519 Issue 4**

41520       Aligned with the ISO/IEC 9945-2: 1993 standard.

41521 **NAME**41522           what — identify SCCS files (**DEVELOPMENT**)41523 **SYNOPSIS**41524 XSI        what [-s] *file...*

41525

41526 **DESCRIPTION**

41527       The *what* utility shall search the given files for all occurrences of the pattern that *get* (see *get* on  
 41528       page 510) substitutes for %Z% ("@( # ) ") and shall write to standard output what follows until  
 41529       the first occurrence of one of the following:

41530           "     &gt;     newline     \     NUL

41531 **OPTIONS**

41532       The *what* utility shall conform to the System Interface Definitions volume of  
 41533       IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

41534       The following option is supported:

41535       -s           Quit after finding the first occurrence of the pattern in each file.

41536 **OPERANDS**

41537       The following operands shall be supported:

41538       *file*           A path name of a file to search.41539 **STDIN**

41540       Not used.

41541 **INPUT FILES**

41542       The input files are of any file type.

41543 **ENVIRONMENT VARIABLES**41544       The following environment variables shall affect the execution of *what*:

41545       *LANG*           Provide a default value for the internationalization variables that are unset or null.  
 41546                       If *LANG* is unset or null, the corresponding value from the implementation-  
 41547                       dependent default locale shall be used. If any of the internationalization variables  
 41548                       contains an invalid setting, the utility shall behave as if none of the variables had  
 41549                       been defined.

41550       *LC\_ALL*          If set to a non-empty string value, override the values of all the other  
 41551                       internationalization variables.

41552       *LC\_CTYPE*       Determine the locale for the interpretation of sequences of bytes of text data as  
 41553                       characters (for example, single-byte as opposed to multi-byte characters in  
 41554                       arguments and input files).

41555       *LC\_MESSAGES*   Determine the locale that should be used to affect the format and contents of  
 41556                       diagnostic messages written to standard error.  
 41557

41558       *NLSPATH*       Determine the location of message catalogs for the processing of *LC\_MESSAGES*.41559 **ASYNCHRONOUS EVENTS**

41560       Default.

41561 **STDOUT**

41562           The standard output shall consist of the following for each *file* operand:

41563           "%s:\n\t%s\n", <pathname>, <identification string>

41564 **STDERR**

41565           Used only for diagnostic messages.

41566 **OUTPUT FILES**

41567           None.

41568 **EXTENDED DESCRIPTION**

41569           None.

41570 **EXIT STATUS**

41571           The following exit values shall be returned:

41572           0    Any matches were found.

41573           1    Otherwise.

41574 **CONSEQUENCES OF ERRORS**

41575           Default.

41576 **APPLICATION USAGE**

41577           The *what* utility is intended to be used in conjunction with the SCCS command *get*, which automatically inserts identifying information, but it can also be used where the information is inserted by any other means.

41580           When the string "@(#)" is included in a library routine in a shared library, it might not be found in an **a.out** file using that library routine.

41582 **EXAMPLES**

41583           If the C-language program in file **f.c** contains:

41584           char ident[] = "@(#)identification information";

41585           and **f.c** is compiled to yield **f.o** and **a.out**, then the command:

41586           what f.c f.o a.out

41587           writes:

41588           f.c:

41589               identification information

41590               ...

41591           f.o:

41592               identification information

41593               ...

41594           a.out:

41595               identification information

41596               ...

41597 **RATIONALE**

41598           None.

41599 **FUTURE DIRECTIONS**

41600           None.



41601 **SEE ALSO**41602 *get*41603 **CHANGE HISTORY**

41604 First released in Issue 2.

41605 **Issue 4**

41606 Format reorganized.

41607 Utility Syntax Guidelines support mandated.

41608 Internationalized environment variable support mandated.

## 41609 NAME

41610       who — display who is on the system

## 41611 SYNOPSIS

41612 UP       who [-mTu]

41613

41614 XSI       who [-mu]-s[-bHlprt][*file*]

41615       who [-mTu][-abdHlprt][*file*]

41616       who -q [*file*]

41617       who am i

41618       who am I

41619

## 41620 DESCRIPTION

41621       The *who* utility shall list various pieces of information about accessible users. The domain of  
41622 accessibility is implementation-dependent.

41623 XSI       Based on the options given, *who* can also list the user's name, terminal line, login time, elapsed  
41624 time since activity occurred on the line, and the process ID of the command interpreter for each  
41625 current system user.

## 41626 OPTIONS

41627       The *who* utility shall conform to the System Interface Definitions volume of  
41628 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

41629       The following options shall be supported. The metavariables, such as *<line>*, refer to fields  
41630 described in the STDOUT section.

41631 XSI       **-a**       Process the implementation-dependent database or named file with the **-b**, **-d**, **-l**,  
41632       **-p**, **-r**, **-t**, **-T** and **-u** options turned on.

41633 XSI       **-b**       Write the time and date of the last reboot.

41634 XSI       **-d**       Write a list of all processes that have expired and not been respawned by the *init*  
41635 system process. The *<exit>* field appears for dead processes and contains the  
41636 termination and exit values of the dead process. This can be useful in determining  
41637 why a process terminated.

41638 XSI       **-H**       Write column headings above the regular output.

41639 XSI       **-l**       (The letter ell.) List only those lines on which the system is waiting for someone to  
41640 login. The *<name>* field is **LOGIN** in such cases. Other fields are the same as for  
41641 user entries except that the *<state>* field does not exist.

41642       **-m**       Output only information about the current terminal.

41643 XSI       **-p**       List any other process that is currently active and has been previously spawned by  
41644 *init*.

41645 XSI       **-q**       (Quick.) List only the names and the number of users currently logged on. When  
41646 this option is used, all other options are ignored.

41647 XSI       **-r**       Write the current *run-level* of the *init* process.

41648 XSI       **-s**       List only the *<name>*, *<line>*, and *<time>* fields. This is the default case.

41649 XSI       **-t**       Indicate the last change to the system clock.

41650           **-T**           Show the state of each terminal, as described in the STDOUT section.

41651 XSI       **-u**           This option lists only those users who are currently logged in. Output the user's  
 41652                           "idle time" in addition to any other information. The idle time is the time since  
 41653                           any activity occurred on the user's terminal. The method of determining this is  
 41654 XSI                       unspecified. The *<name>* is the user's login name. The *<line>* is the name of the line  
 41655                           as found in the directory */dev*. The *<time>* is the time that the user logged in. The  
 41656                           *<activity>* is the number of hours and minutes since activity last occurred on that  
 41657                           particular line. A dot indicates that the terminal has seen activity in the last minute  
 41658                           and is therefore "current". If more than twenty-four hours have elapsed or the line  
 41659                           has not been used since boot time, the entry is marked *<old>*. This field is useful  
 41660                           when trying to determine whether a person is working at the terminal or not. The  
 41661                           *<pid>* is the process ID of the user's login process.

#### 41662 OPERANDS

41663 XSI       The following operands shall be supported:

41664           **am i, am I**   In the POSIX locale, limit the output to describing the invoking user, equivalent to  
 41665                           the **-m** option. The **am** and **i** or **I** must be separate arguments.

41666           **file**       Specify a path name of a file to substitute for the implementation-dependent  
 41667                           database of logged-on users that *who* uses by default.

#### 41668 STDIN

41669           Not used.

#### 41670 INPUT FILES

41671           None.

#### 41672 ENVIRONMENT VARIABLES

41673           The following environment variables shall affect the execution of *who*:

41674           **LANG**       Provide a default value for the internationalization variables that are unset or null.  
 41675                           If **LANG** is unset or null, the corresponding value from the implementation-  
 41676                           dependent default locale shall be used. If any of the internationalization variables  
 41677                           contains an invalid setting, the utility shall behave as if none of the variables had  
 41678                           been defined.

41679           **LC\_ALL**       If set to a non-empty string value, override the values of all the other  
 41680                           internationalization variables.

41681           **LC\_CTYPE**   Determine the locale for the interpretation of sequences of bytes of text data as  
 41682                           characters (for example, single-byte as opposed to multi-byte characters in  
 41683                           arguments).

41684           **LC\_MESSAGES**   Determine the locale that should be used to affect the format and contents of  
 41685                           diagnostic messages written to standard error.  
 41686                           

41687           **LC\_TIME**     Determine the locale used for the format and contents of the date and time strings.

41688 XSI       **NLSPATH**   Determine the location of message catalogs for the processing of **LC\_MESSAGES**.

#### 41689 ASYNCHRONOUS EVENTS

41690           Default.

41691 **STDOUT**

41692 XSI OF XSI-conformant systems shall write the default information to the standard output in the  
 41693 following general format:

41694 `<name>[<state>]<line><time>[<activity>][<pid>][<comment>][<exit>]`

41695 The following format shall be used for the **-T** option:

41696 `"%s %c %s %s\n" <name>, <terminal state>, <terminal name>,  
 41697 <time of login>`

41698 where *<terminal state>* is one of the following characters:

- 41699 + The terminal allows write access to other users.
- 41700 – The terminal denies write access to other users.
- 41701 ? The terminal write-access state cannot be determined.

41702 In the POSIX locale, the *<time of login>* shall be equivalent in format to the output of:

41703 `date +"%b %e %H:%M"`

41704 If the **-u** option is used with **-T**, the idle time shall be added to the end of the previous format in  
 41705 an unspecified format.

41706 **STDERR**

41707 Used only for diagnostic messages.

41708 **OUTPUT FILES**

41709 None.

41710 **EXTENDED DESCRIPTION**

41711 None.

41712 **EXIT STATUS**

41713 The following exit values shall be returned:

- 41714 0 Successful completion.
- 41715 >0 An error occurred.

41716 **CONSEQUENCES OF ERRORS**

41717 Default.

41718 **APPLICATION USAGE**

41719 The name *init* used for the system process is the most commonly used on historical systems, but  
 41720 it may vary.

41721 The “domain of accessibility” referred to is a broad concept that permits interpretation either on  
 41722 a very secure basis or even to allow a network-wide implementation like the historical *rwwho*.

41723 Application writers should note that this utility need not be provided on systems that do not  
 41724 support the User Portability Utilities option.

41725 **EXAMPLES**

41726 None.

41727 **RATIONALE**

41728 Due to differences between historical implementations, the base options provided were a  
 41729 compromise to allow users to work with those functions. The standard developers also  
 41730 considered removing all the options, but felt that these options offered users valuable  
 41731 functionality. Additional options to match historical systems are available on XSI-conformant

- 41732 systems.
- 41733 It is recognized that the *who* command may be of limited usefulness, especially in a multi-level  
41734 secure environment. The standard developers considered, however, that having some standard  
41735 method of determining the “accessibility” of other users would aid user portability.
- 41736 No format was specified for the default *who* output for systems not supporting the XSI  
41737 Extension. In such a user-oriented command, designed only for human use, this was not  
41738 considered to be a deficiency.
- 41739 The format of the terminal name is unspecified, but the descriptions of *ps*, *talk*, and *write* require  
41740 that they use the same format.
- 41741 **FUTURE DIRECTIONS**
- 41742 None.
- 41743 **SEE ALSO**
- 41744 *mesg*
- 41745 **CHANGE HISTORY**
- 41746 First released in Issue 2.
- 41747 **Issue 4**
- 41748 Aligned with the ISO/IEC 9945-2: 1993 standard.
- 41749 **Issue 6**
- 41750 This utility is now marked as part of the User Portability Utilities option.

## 41751 NAME

41752 write — write to another user

## 41753 SYNOPSIS

41754 UP write *user\_name* [*terminal*]

41755

## 41756 DESCRIPTION

41757 The *write* utility shall read lines from the user's standard input and write them to the terminal of  
 41758 another user. When first invoked, it shall write the message:

41759 **Message from** *sender-login-id* (*sending-terminal*) [*date*]...

41760 to *user\_name*. When it has successfully completed the connection, the sender's terminal shall be  
 41761 alerted twice to indicate that what the sender is typing is being written to the recipient's  
 41762 terminal.

41763 If the recipient wants to reply, this can be accomplished by typing:

41764 write *sender-login-id* [*sending-terminal*]

41765 upon receipt of the initial message. Whenever a line of input as delimited by a NL, EOF, or EOL  
 41766 special character (see the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter  
 41767 11, General Terminal Interface) is accumulated while in canonical input mode, the accumulated  
 41768 data shall be written on the other user's terminal. Characters shall be processed as follows:

- 41769 • Typing the <alert> character shall write the alert character to the recipient's terminal.
- 41770 • Typing the erase and kill characters shall affect the sender's terminal in the manner described  
 41771 by the **termios** interface in the System Interface Definitions volume of IEEE Std. 1003.1-200x,  
 41772 Chapter 11, General Terminal Interface.
- 41773 • Typing the interrupt or end-of-file characters shall cause *write* to write an appropriate  
 41774 message ("EOT\n" in the POSIX locale) to the recipient's terminal and exit.
- 41775 • Typing characters from *LC\_CTYPE* classifications **print** or **space** shall cause those characters  
 41776 to be sent to the recipient's terminal.
- 41777 • When and only when the *stty ixten* local mode is enabled, the existence and processing of  
 41778 additional special control characters and multi-byte or single-byte functions is  
 41779 implementation-dependent.
- 41780 • Typing other non-printable characters shall cause implementation-dependent sequences of  
 41781 printable characters to be written to the recipient's terminal.

41782 To write to a user who is logged in more than once, the *terminal* argument can be used to indicate  
 41783 which terminal to write to; otherwise, the recipient's terminal is selected in an implementation-  
 41784 dependent manner and an informational message is written to the sender's standard output,  
 41785 indicating which terminal was chosen.

41786 Permission to be a recipient of a *write* message can be denied or granted by use of the *mesg*  
 41787 utility. However, a user's privilege may further constrain the domain of accessibility of other  
 41788 users' terminals. The *write* utility shall fail when the user lacks the appropriate privileges to  
 41789 perform the requested action.

## 41790 OPTIONS

41791 None.

41792 **OPERANDS**

41793 The following operands shall be supported:

41794 *user\_name* Login name of the person to whom the message shall be written. The application  
 41795 shall ensure that this operand is of the form returned by the *who* utility.

41796 *terminal* Terminal identification in the same format provided by the *who* utility.

41797 **STDIN**

41798 Lines to be copied to the recipient's terminal is read from standard input.

41799 **INPUT FILES**

41800 None.

41801 **ENVIRONMENT VARIABLES**41802 The following environment variables shall affect the execution of *write*:

41803 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 41804 If *LANG* is unset or null, the corresponding value from the implementation-  
 41805 dependent default locale shall be used. If any of the internationalization variables  
 41806 contains an invalid setting, the utility shall behave as if none of the variables had  
 41807 been defined.

41808 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 41809 internationalization variables.

41810 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 41811 characters (for example, single-byte as opposed to multi-byte characters in  
 41812 arguments and input files). If the recipient's locale does not use an *LC\_CTYPE*  
 41813 equivalent to the sender's, the results are undefined.

41814 *LC\_MESSAGES*

41815 Determine the locale that should be used to affect the format and contents of  
 41816 diagnostic messages written to standard error and informative messages written to  
 41817 standard output.

41818 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

41819 **ASYNCHRONOUS EVENTS**

41820 If an interrupt signal is received, *write* shall write an appropriate message on the recipient's  
 41821 terminal and exits with a status of zero. It shall take the standard action for all other signals.

41822 **STDOUT**

41823 An informational message shall be written to standard output if a recipient is logged in more  
 41824 than once.

41825 **STDERR**

41826 Used only for diagnostic messages.

41827 **OUTPUT FILES**

41828 The recipient's terminal is used for output.

41829 **EXTENDED DESCRIPTION**

41830 None.

41831 **EXIT STATUS**

41832 The following exit values shall be returned:

41833 0 Successful completion.

41834 >0 The addressed user is not logged on or the addressed user denies permission.

41835 **CONSEQUENCES OF ERRORS**

41836 Default.

41837 **APPLICATION USAGE**

41838 The *talk* utility is considered by some users to be a more usable utility on full-screen terminals.

41839 Application writers should note that this utility need not be provided on systems that do not  
41840 support the User Portability Utilities option.

41841 **EXAMPLES**

41842 None.

41843 **RATIONALE**

41844 The *write* utility was included in this volume of IEEE Std. 1003.1-200x since it can be  
41845 implemented on all terminal types. The standard developers considered the *talk* utility, which  
41846 cannot be implemented on certain terminals, to be a “better” communications interface. Both of  
41847 these programs are in widespread use on historical implementations. Therefore, the standard  
41848 developers decided that both utilities should be specified.

41849 The format of the terminal name is unspecified, but the descriptions of *ps*, *talk*, *who*, and *write*  
41850 require that they all use or accept the same format.

41851 **FUTURE DIRECTIONS**

41852 None.

41853 **SEE ALSO**

41854 *mesg*, *talk*, *who*, the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 11,  
41855 General Terminal Interface

41856 **CHANGE HISTORY**

41857 First released in Issue 2.

41858 **Issue 4**

41859 Aligned with the ISO/IEC 9945-2: 1993 standard.

41860 **Issue 5**

41861 FUTURE DIRECTIONS section added.

41862 **Issue 6**

41863 This utility is now marked as part of the User Portability Utilities option.

41864 The normative text is reworded to avoid use of the term “must” for application requirements.



41865 **NAME**41866 **xargs** — construct argument lists and invoke utility41867 **SYNOPSIS**

```
41868 xsi xargs [-t][-p][-E eofstr][-I replstr][-L number][-n number [-x]]
```

```
41869 [-s size][utility [argument...]]
```

41870 **DESCRIPTION**

41871 The *xargs* utility shall construct a command line consisting of the *utility* and *argument* operands  
 41872 specified followed by as many arguments read in sequence from standard input as fit in length  
 41873 and number constraints specified by the options. The *xargs* utility shall then invoke the  
 41874 constructed command line and wait for its completion. This sequence shall be repeated until one  
 41875 of the following occurs:

- 41876 • An end-of-file condition is detected on standard input.
- 41877 • The logical end-of-file string (see the *-E eofstr* option) is found on standard input after  
 41878 double-quote processing, apostrophe processing, and backslash escape processing (see next  
 41879 paragraph).
- 41880 • An invocation of a constructed command line returns an exit status of 255.

41881 The application shall ensure that arguments in the standard input are separated by unquoted  
 41882 <blank> characters, or unescaped <blank> characters or <newline> characters. A string of zero  
 41883 or more non-double-quote ( ' ) ' and non-<newline> characters can be quoted by enclosing  
 41884 them in double-quotes. A string of zero or more non-apostrophe ( ' ' ' ) and non-<newline>  
 41885 characters can be quoted by enclosing them in apostrophes. Any unquoted character can be  
 41886 escaped by preceding it with a backslash. The utility shall be executed one or more times until  
 41887 the end-of-file is reached or the logical end-of file string is found. The results are unspecified if  
 41888 the utility named by *utility* attempts to read from its standard input.

41889 The generated command line length shall be the sum of the size in bytes of the utility name and  
 41890 each argument treated as strings, including a null byte terminator for each of these strings. The  
 41891 *xargs* utility shall limit the command line length such that when the command line is invoked,  
 41892 the combined argument and environment lists (see the *exec* family of functions in the System  
 41893 Interfaces volume of IEEE Std. 1003.1-200x) shall not exceed {ARG\_MAX}-2 048 bytes. Within  
 41894 this constraint, if neither the *-n* nor the *-s* option is specified, the default command line length  
 41895 shall be at least {LINE\_MAX}.

41896 **OPTIONS**

41897 The *xargs* utility shall conform to the System Interface Definitions volume of  
 41898 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

41899 The following options shall be supported:

41900 xsi ***-E eofstr*** Use *eofstr* as the logical end-of-file string. If *-E* is not specified, it is unspecified  
 41901 whether the logical end-of-file string is the underscore character ( ' \_ ' ) or the end-  
 41902 of-file string capability is disabled. When *eofstr* is the null string, the logical end-  
 41903 of-file string capability shall be disabled and underscore characters shall be taken  
 41904 literally.

41905 xsi ***-I replstr*** Insert mode: *utility* is executed for each line from standard input, taking the entire  
 41906 line as a single argument, inserting it in *arguments* for each occurrence of *replstr*. A  
 41907 maximum of five arguments in *arguments* can each contain one or more instances  
 41908 of *replstr*. Any <blank> characters at the beginning of each line shall be ignored.  
 41909 Constructed arguments cannot grow larger than 255 bytes. Option *-x* is forced on.  
 41910 The *-I* and *-i* options are mutually-exclusive; the last one specified shall take  
 41911 effect.

|           |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|-----------|--------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 41912 XSI | <b>-L number</b>                           | The <i>utility</i> shall be executed for each non-empty <i>number</i> lines of arguments from standard input. The last invocation of <i>utility</i> shall be with fewer lines of arguments if fewer than <i>number</i> remain. A line is considered to end with the first <newline> character unless the last character of the line is a <blank> character; a trailing <blank> character signals continuation to the next non-empty line, inclusive. The <b>-L</b> , <b>-l</b> , and <b>-n</b> options are mutually-exclusive; the last one specified shall take effect. |
| 41913     |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41914     |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41915     |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41916     |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41917     |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41918     |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41919     | <b>-n number</b>                           | Invoke <i>utility</i> using as many standard input arguments as possible, up to <i>number</i> (a positive decimal integer) arguments maximum. Fewer arguments shall be used if:                                                                                                                                                                                                                                                                                                                                                                                          |
| 41920     |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41921     |                                            | <ul style="list-style-type: none"> <li>• The command line length accumulated exceeds the size specified by the <b>-s</b> option (or {LINE_MAX} if there is no <b>-s</b> option).</li> </ul>                                                                                                                                                                                                                                                                                                                                                                              |
| 41922     |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41923     |                                            | <ul style="list-style-type: none"> <li>• The last iteration has fewer than but not zero, operands remaining.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 41924 MAN | <b>-p</b>                                  | Prompt mode: the user is asked whether to execute <i>utility</i> at each invocation. Trace mode ( <b>-t</b> ) is turned on to write the command instance to be executed, followed by a prompt to standard error. An affirmative response read from <i>/dev/tty</i> shall execute the command; otherwise, that particular invocation of <i>utility</i> shall be skipped.                                                                                                                                                                                                  |
| 41925     |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41926     |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41927     |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41928     |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41929     | <b>-s size</b>                             | Invoke <i>utility</i> using as many standard input arguments as possible yielding a command line length less than <i>size</i> (a positive decimal integer) bytes. Fewer arguments shall be used if:                                                                                                                                                                                                                                                                                                                                                                      |
| 41930     |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41931     |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41932     |                                            | <ul style="list-style-type: none"> <li>• The total number of arguments exceeds that specified by the <b>-n</b> option.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 41933 XSI |                                            | <ul style="list-style-type: none"> <li>• The total number of lines exceeds that specified by the <b>-L</b> option.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 41934     |                                            | <ul style="list-style-type: none"> <li>• End-of-file is encountered on standard input before <i>size</i> bytes are accumulated.</li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 41935     |                                            | Values of <i>size</i> up to at least {LINE_MAX} bytes shall be supported, provided that the constraints specified in the DESCRIPTION are met. It shall not be considered an error if a value larger than that supported by the implementation or exceeding the constraints specified in the DESCRIPTION is given; <i>xargs</i> shall use the largest value it supports within the constraints.                                                                                                                                                                           |
| 41936     |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41937     |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41938     |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41939     |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41940     | <b>-t</b>                                  | Enable trace mode. Each generated command line shall be written to standard error just prior to invocation.                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| 41941     |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41942     | <b>-x</b>                                  | Terminate if a command line containing <i>number</i> arguments (see the <b>-n</b> option above) or <i>number</i> lines (see the <b>-L</b> option above) will not fit in the implied or specified size (see the <b>-s</b> option above).                                                                                                                                                                                                                                                                                                                                  |
| 41943 XSI |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41944     |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41945     | <b>OPERANDS</b>                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41946     | The following operands shall be supported: |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41947     | <i>utility</i>                             | The name of the utility to be invoked, found by search path using the <i>PATH</i> environment variable, described in the System Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 8, Environment Variables. If <i>utility</i> is omitted, the default shall be the <i>echo</i> utility. If the <i>utility</i> operand names any of the special built-in utilities in Section 2.14 on page 96, the results are undefined.                                                                                                                                    |
| 41948     |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41949     |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41950     |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41951     |                                            |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 41952     | <i>argument</i>                            | An initial option or operand for the invocation of <i>utility</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |

41953 **STDIN**

41954 The standard input shall be a text file. The results are unspecified if an end-of-file condition is  
 41955 detected immediately following an escaped <newline> character.

41956 **INPUT FILES**

41957 MAN The file `/dev/tty` is used to read responses required by the `-p` option.

41958 **ENVIRONMENT VARIABLES**

41959 The following environment variables shall affect the execution of *xargs*:

41960 *LANG* Provide a default value for the internationalization variables that are unset or null.  
 41961 If *LANG* is unset or null, the corresponding value from the implementation-  
 41962 dependent default locale shall be used. If any of the internationalization variables  
 41963 contains an invalid setting, the utility shall behave as if none of the variables had  
 41964 been defined.

41965 *LC\_ALL* If set to a non-empty string value, override the values of all the other  
 41966 internationalization variables.

41967 *LC\_COLLATE*

41968 Determine the locale for the behavior of ranges, equivalence classes and multi-  
 41969 character collating elements used in the extended regular expression defined for  
 41970 the **yesexpr** locale keyword in the *LC\_MESSAGES* category.

41971 *LC\_CTYPE* Determine the locale for the interpretation of sequences of bytes of text data as  
 41972 characters (for example, single-byte as opposed to multi-byte characters in  
 41973 arguments and input files) and the behavior of character classes used in the  
 41974 extended regular expression defined for the **yesexpr** locale keyword in the  
 41975 *LC\_MESSAGES* category.

41976 *LC\_MESSAGES*

41977 Determine the locale for the processing of affirmative responses and that should be  
 41978 used to affect the format and contents of diagnostic messages written to standard  
 41979 error.

41980 XSI *NLSPATH* Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

41981 *PATH* Determine the location of *utility*, as described in the System Interface Definitions  
 41982 volume of IEEE Std. 1003.1-200x, Chapter 8, Environment Variables.

41983 **ASYNCHRONOUS EVENTS**

41984 Default.

41985 **STDOUT**

41986 Not used.

41987 **STDERR**

41988 MAN Used for diagnostic messages and the `-t` and `-p` options. If the `-t` option is specified, the *utility*  
 41989 and its constructed argument list shall be written to standard error, as it will be invoked, prior to  
 41990 MAN invocation. If `-p` is specified, a prompt of the following format shall be written (in the POSIX  
 41991 locale):

41992 " . . . "

41993 at the end of the line of the output from `-t`.

41994 **OUTPUT FILES**

41995       None.

41996 **EXTENDED DESCRIPTION**

41997       None.

41998 **EXIT STATUS**

41999       The following exit values shall be returned:

- 42000           0    All invocations of *utility* returned exit status zero.
- 42001           1-125   A command line meeting the specified requirements could not be assembled, one or more of the invocations of *utility* returned a non-zero exit status, or some other error occurred.
- 42002
- 42003
- 42004           126   The utility specified by *utility* was found but could not be invoked.
- 42005           127   The utility specified by *utility* could not be found.

42006 **CONSEQUENCES OF ERRORS**

42007       If a command line meeting the specified requirements cannot be assembled, the utility cannot be invoked, an invocation of the utility is terminated by a signal, or an invocation of the utility exits with exit status 255, the *xargs* utility shall write a diagnostic message and exit without processing any remaining input.

42008

42009

42010

42011 **APPLICATION USAGE**

42012       The 255 exit status allows a utility being used by *xargs* to tell *xargs* to terminate if it knows no further invocations using the current data stream succeeds. Thus, *utility* should explicitly *exit* with an appropriate value to avoid accidentally returning with 255.

42013

42014

42015       Note that input is parsed as lines; <blank> characters separate arguments. If *xargs* is used to bundle output of commands like *find dir -print* or *ls* into commands to be executed, unexpected results are likely if any file names contain any <blank> characters or <newline> characters. This can be fixed by using *find* to call a script that converts each file found into a quoted string that is then piped to *xargs*. Note that the quoting rules used by *xargs* are not the same as in the shell. They were not made consistent here because existing applications depend on the current rules and the shell syntax is not fully compatible with it. An easy rule that can be used to transform any string into a quoted form that *xargs* interprets correctly is to precede each character in the string with a backslash.

42016

42017

42018

42019

42020

42021

42022

42023

42024       On implementations with a large value for {ARG\_MAX}, *xargs* may produce command lines longer than {LINE\_MAX}. For invocation of utilities, this is not a problem. If *xargs* is being used to create a text file, users should explicitly set the maximum command line length with the *-s* option.

42025

42026

42027

42028       The *command*, *env*, *nice*, *nohup*, *time*, and *xargs* utilities have been specified to use exit code 127 if an error occurs so that applications can distinguish “failure to find a utility” from “invoked utility exited with an error indication”. The value 127 was chosen because it is not commonly used for other meanings; most utilities use small values for “normal error conditions” and the values above 128 can be confused with termination due to receipt of a signal. The value 126 was chosen in a similar manner to indicate that the utility could be found, but not invoked. Some scripts produce meaningful error messages differentiating the 126 and 127 cases. The distinction between exit codes 126 and 127 is based on KornShell practice that uses 127 when all attempts to *exec* the utility fail with [ENOENT], and uses 126 when any attempt to *exec* the utility fails for any other reason.

42029

42030

42031

42032

42033

42034

42035

42036

42037

## 42038 EXAMPLES

42039 1. The following moves all files from directory **\$1** to directory **\$2**, and echo each move  
 42040 command just before doing it:

42041 `ls $1 | xargs -I {} -t mv $1/{} $2/{} |`

42042 2. The following command combines the output of the parenthesised commands onto one  
 42043 line, which is then written to the end-of-file **log**:

42044 `(logname; date; printf "%s\n" "$0 $*") | xargs >>log`

42045 3. The following command invokes *diff* with successive pairs of arguments originally typed  
 42046 as command line arguments (assuming there are no embedded <blank> characters in the  
 42047 elements of the original argument list):

42048 `printf "%s\n" "$*" | xargs -n 2 -x diff`

42049 4. The user is asked which files in the current directory shall be archived. The files are  
 42050 archived into **arch**; *a*, one at a time, or *b*, many at a time.

42051 `a. ls | xargs -p -L 1 ar -r arch`

42052 `b. ls | xargs -p -L 1 | xargs ar -r arch`

42053 5. The following executes with successive pairs of arguments originally typed as command  
 42054 line arguments:

42055 `echo $* | xargs -n 2 diff`

## 42056 RATIONALE

42057 The *xargs* utility was usually found only in System V-based systems; BSD systems included an  
 42058 *apply* utility that provided functionality similar to *xargs -n number*. The SVID lists *xargs* as a  
 42059 software development extension. This volume of IEEE Std. 1003.1-200x does not share the view  
 42060 that it is used only for development, and therefore it is not optional.

42061 The classic application of the *xargs* utility is in conjunction with the *find* utility to reduce the  
 42062 number of processes launched by a simplistic use of the *find-exec* combination. The *xargs* utility  
 42063 is also used to enforce an upper limit on memory required to launch a process. With this basis in  
 42064 mind, this volume of IEEE Std. 1003.1-200x selected only the minimal features required.

42065 The *-n number* option was used classically to evoke a utility using pairs of operands, yet the  
 42066 general case has problems when *utility* spawns child processes of its own. The *xargs* utility can  
 42067 sap resources from these children, especially those sharing the environment of the parent.

42068 The *command*, *env*, *nohup*, and *xargs* utilities have been specified to use exit code 127 if an error  
 42069 occurs so that applications can distinguish “failure to find a utility” from “invoked utility exited  
 42070 with an error indication”. The value 127 was chosen because it is not commonly used for other  
 42071 meanings; most utilities use small values for “normal error conditions”, and the values above  
 42072 128 can be confused with termination due to receipt of a signal. The value 126 was chosen in a  
 42073 similar manner to indicate that the utility could be found, but not invoked. Some scripts produce  
 42074 meaningful error messages differentiating the 126 and 127 cases. The distinction between exit  
 42075 codes 126 and 127 is based on KornShell practice that uses 127 when all attempts to *exec* the  
 42076 utility fail with [ENOENT] and that uses 126 when any attempt to *exec* the utility fails for any  
 42077 other reason.

42078 Although the 255 exit status is mostly an accident of historical implementations, it allows a  
 42079 utility being used by *xargs* to tell *xargs* to terminate if it knows no further invocations using the  
 42080 current data stream shall succeed. Any non-zero exit status from a utility falls into the 1-125  
 42081 range when *xargs* exits. There is no statement of how the various non-zero utility exit status

codes are accumulated by *xargs*. The value could be the addition of all codes, their highest value, the last one received, or a single value such as 1. Since no algorithm is arguably better than the others, and since many of the standard utilities say little more (portably) than “pass/fail”, no new algorithm was invented.

Several other *xargs* options were withdrawn because simple alternatives already exist within this volume of IEEE Std. 1003.1-200x. For example, the *-e eofstr* option can be replaced by features of *sed*. The *-i replstr* option can be just as efficiently performed using a shell *for* loop. Since *xargs* calls an *exec* function with each input line, the *-i* option does not usually exploit the grouping capabilities of *xargs*.

The requirement that *xargs* never produce command lines such that invocation of *utility* is within 2 048 bytes of hitting the POSIX *exec* {ARG\_MAX} limitations is intended to guarantee that the invoked utility has room to modify its environment variables and command line arguments and still be able to invoke another utility. Note that the minimum {ARG\_MAX} allowed by the System Interfaces volume of IEEE Std. 1003.1-200x is 4 096 bytes and the minimum value allowed by the this volume of IEEE Std. 1003.1-200x is 2 048 bytes; therefore, the 2 048 bytes difference seems reasonable. Note, however, that *xargs* may never be able to invoke a utility if the environment passed in to *xargs* comes close to using {ARG\_MAX} bytes.

The version of *xargs* required by this volume of IEEE Std. 1003.1-200x is required to wait for the completion of the invoked command before invoking another command. This was done because historical scripts using *xargs* assumed sequential execution. Implementations wanting to provide parallel operation of the invoked utilities are encouraged to add an option enabling parallel invocation, but should still wait for termination of all of the children before *xargs* terminates normally.

The *-e* option was omitted from the ISO POSIX-2:1993 standard in the belief that the *eofstr* option-argument was recognized only when it was on a line by itself and before quote and escape processing were performed, and that the logical end-of-file processing was only enabled if a *-e* option was specified. In that case, a simple *sed* script could be used to duplicate the *-e* functionality. Further investigation revealed that:

- The logical end-of-file string was checked for after quote and escape processing, making a *sed* script that provided equivalent functionality much more difficult to write.
- The default was to perform logical end-of-file processing with an underscore as the logical end-of-file string.

To correct this misunderstanding, the *-E eofstr* option was adopted from Issue 4 in an amendment to IEEE Std. 1003.1-200x. Users should note that the description of the *-E* option matches historical documentation of the *-e* option (which was not adopted because it did not support the Utility Syntax Guidelines), by saying that if *eofstr* is the null string, logical end-of-file processing is disabled. Historical implementations of *xargs* actually did not disable logical end-of-file processing; they treated a null argument found in the input as a logical end-of-file string. (A null *string* argument could be generated using single or double quotes ( ' ' or " "). Since this behavior was not documented historically, it is considered to be a bug.

## FUTURE DIRECTIONS

A version supporting the Utility Syntax Guidelines may be introduced.

## SEE ALSO

*echo*

42126 **CHANGE HISTORY**

42127 First released in Issue 2.

42128 **Issue 4**

42129 Aligned with the ISO/IEC 9945-2: 1993 standard.

42130 **Issue 5**

42131 Second FUTURE DIRECTION added.

42132 **Issue 6**42133 The obsolescent **-e**, **-i**, and **-l** options are removed.42134 The following new requirements on POSIX implementations derive from alignment with the  
42135 Single UNIX Specification:

- 42136 • The **-p** option is added.
- 42137 • In the INPUT FILES section, the file **/dev/tty** is used to read responses required by the **-p**  
42138 option.
- 42139 • The STDERR section is updated to describe the **-p** option.

42140 The description of the **-E** option is aligned with the ISO POSIX-2: 1993 standard. |

42141 The normative text is reworded to avoid use of the term “must” for application requirements. |

## 42142 NAME

42143 yacc — yet another compiler compiler (DEVELOPMENT)

## 42144 SYNOPSIS

42145 yacc [-dltv][*-b file\_prefix*][*-p sym\_prefix*] *grammar*

## 42146 DESCRIPTION

42147 The *yacc* utility shall read a description of a context-free grammar in *file* and write C source code,  
 42148 conforming to the ISO C standard, to a code file, and optionally header information into a  
 42149 header file, in the current directory. The C code shall define a function and related routines and  
 42150 macros for an automaton that executes a parsing algorithm meeting the requirements in  
 42151 **Algorithms** on page 1119.

42152 The form and meaning of the grammar are described in the EXTENDED DESCRIPTION section.

42153 The C source code and header file shall be produced in a form suitable as input for the C  
 42154 compiler (see *c89* on page 246).

## 42155 OPTIONS

42156 The *yacc* utility shall conform to the System Interface Definitions volume of  
 42157 IEEE Std. 1003.1-200x, Section 12.2, Utility Syntax Guidelines.

42158 The following options shall be supported:

42159 **-b *file\_prefix*** Use *file\_prefix* instead of *y* as the prefix for all output file names. The code file  
 42160 *y.tab.c*, the header file *y.tab.h* (created when **-d** is specified), and the description  
 42161 file *y.output* (created when **-v** is specified), shall be changed to *file\_prefix.tab.c*,  
 42162 *file\_prefix.tab.h*, and *file\_prefix.output*, respectively.

42163 **-d** Write the header file; by default only the code file is written. The **#define**  
 42164 statements that associate the token codes assigned by *yacc* with the user-declared  
 42165 token names. This allows source files other than *y.tab.c* to access the token codes.

42166 **-l** Produce a code file that does not contain any **#line** constructs. If this option is not  
 42167 present, it is unspecified whether the code file or header file contains **#line**  
 42168 directives. This should only be used after the grammar and the associated actions  
 42169 are fully debugged.

42170 **-p *sym\_prefix*** Use *sym\_prefix* instead of *yy* as the prefix for all external names produced by *yacc*.  
 42171 The names affected shall include the functions *yyparse*, *yylex*, and *yyerror*, and the  
 42172 variables *yylval*, *yychar*, and *yydebug*. (In the remainder of this section, the six  
 42173 symbols cited are referenced using their default names only as a notational  
 42174 convenience.) Local names may also be affected by the **-p** option; however, the **-p**  
 42175 option shall not affect **#define** symbols generated by *yacc*.

42176 **-t** Modify conditional compilation directives to permit compilation of debugging  
 42177 code in the code file. Runtime debugging statements shall always be contained in  
 42178 the code file, but by default conditional compilation directives prevent their  
 42179 compilation.

42180 **-v** Write a file containing a description of the parser and a report of conflicts  
 42181 generated by ambiguities in the grammar.

## 42182 OPERANDS

42183 The following operand is required:

42184 *grammar* A path name of a file containing instructions, hereafter called *grammar*, for which a  
 42185 parser is to be created. The format for the grammar is described in the EXTENDED  
 42186 DESCRIPTION section.



**42187 STDIN**

42188 Not used.

**42189 INPUT FILES**

42190 The file *grammar* shall be a text file formatted as specified in the EXTENDED DESCRIPTION  
42191 section.

**42192 ENVIRONMENT VARIABLES**

42193 The following environment variables shall affect the execution of *yacc*:

42194 **LANG** Provide a default value for the internationalization variables that are unset or null.  
42195 If *LANG* is unset or null, the corresponding value from the implementation-  
42196 dependent default locale shall be used. If any of the internationalization variables  
42197 contains an invalid setting, the utility shall behave as if none of the variables had  
42198 been defined.

42199 **LC\_ALL** If set to a non-empty string value, override the values of all the other  
42200 internationalization variables.

42201 **LC\_CTYPE** Determine the locale for the interpretation of sequences of bytes of text data as  
42202 characters (for example, single-byte as opposed to multi-byte characters in  
42203 arguments and input files).

**42204 LC\_MESSAGES**

42205 Determine the locale that should be used to affect the format and contents of  
42206 diagnostic messages written to standard error.

42207 XSI **NLSPATH** Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

42208 The *LANG* and *LC\_\** variables affect the execution of the *yacc* utility as stated. The *main* function  
42209 defined in **Yacc Library** on page 1119 shall call:

42210 `setlocale(LC_ALL, " ")`

42211 and thus, the program generated by *yacc* also shall be affected by the contents of these variables  
42212 at runtime.

**42213 ASYNCHRONOUS EVENTS**

42214 Default.

**42215 STDOUT**

42216 Not used.

**42217 STDERR**

42218 If shift/reduce or reduce/reduce conflicts are detected in *grammar*, *yacc* writes a report of those  
42219 conflicts to the standard error in an unspecified format.

42220 Standard error is also used for diagnostic messages.

**42221 OUTPUT FILES**

42222 The code file, the header file, and the description file shall be text files. All are described in the  
42223 following sections.

42224      **Code File**

42225      This file shall contain the C source code for the *yyparse* routine. It shall contain code for the  
 42226      various semantic actions with macro substitution performed on them as described in the  
 42227      EXTENDED DESCRIPTION section. It also shall contain a copy of the **#define** statements in the  
 42228      header file. If a **%union** declaration is used, the declaration for YYSTYPE shall be also included  
 42229      in this file.

42230      **Header File**

42231      The header file shall contain **#define** statements that associate the token numbers with the token  
 42232      names. This allows source files other than the code file to access the token codes. If a **%union**  
 42233      declaration is used, the declaration for YYSTYPE and an *extern YYSTYPE yylval* declaration shall  
 42234      be also included in this file.

42235      **Description File**

42236      The description file shall be a text file containing a description of the state machine  
 42237      corresponding to the parser, using an unspecified format. Limits for internal tables (see **Limits**  
 42238      on page 1120) shall also be reported, in an implementation-dependent manner. (Some  
 42239      implementations may use dynamic allocation techniques and have no specific limit values to  
 42240      report.)

42241      **EXTENDED DESCRIPTION**

42242      The *yacc* command accepts a language that is used to define a grammar for a target language to  
 42243      be parsed by the tables and code generated by *yacc*. The language accepted by *yacc* as a  
 42244      grammar for the target language is described below using the *yacc* input language itself.

42245      The input *grammar* includes rules describing the input structure of the target language and code  
 42246      to be invoked when these rules are recognized to provide the associated semantic action. The  
 42247      code to be executed shall appear as bodies of text that are intended to be C-language code. The  
 42248      C-language inclusions are presumed to form a correct function when processed by *yacc* into its  
 42249      output files. The code included in this way shall be executed during the recognition of the target  
 42250      language.

42251      Given a grammar, the *yacc* utility generates the files described in the OUTPUT FILES section.  
 42252      The code file can be compiled and linked using *cc* or *c89*. If the declaration and programs  
 42253      sections of the grammar file did not include definitions of *main*, *yylex*, and *yyerror*, the compiled  
 42254      output requires linking with externally supplied version of those functions. Default versions of  
 42255      *main* and *yyerror* are supplied in the *yacc* library and can be linked in by using the *-ly* operand to  
 42256      *c89*. The *yacc* library interfaces need not support interfaces with other than the default **yy**  
 42257      symbol prefix. The application provides the lexical analyzer function, *yylex*; the *lex* utility is  
 42258      specifically designed to generate such a routine.

42259      **Input Language**

42260      The application ensure that every specification file consists of three sections in order:  
 42261      *declarations*, *grammar rules*, and *programs*, separated by double percent signs ("%"). The  
 42262      declarations and programs sections can be empty. If the latter is empty, the preceding "%"   
 42263      mark separating it from the rules section can be omitted.

42264      The input is free form text following the structure of the grammar defined below.

42265      **Lexical Structure of the Grammar**

42266      The characters <blank>, <newline>, and <form-feed> shall be ignored, except that the  
 42267      application shall ensure that they do not appear in names or multi-character reserved symbols.  
 42268      Comments shall be enclosed in `"/* . . . */"`, and can appear wherever a name is valid.

42269      Names are of arbitrary length, made up of letters, periods (`'.'`), underscores (`'_'`), and non-  
 42270      initial digits. Uppercase and lowercase letters are distinct. Portable applications shall not use  
 42271      names beginning in `yy` or `YY` since the `yacc` parser uses such names. Many of the names appear  
 42272      in the final output of `yacc`, and thus they should be chosen to conform with any additional rules  
 42273      created by the C compiler to be used. In particular they appear in `#define` statements.

42274      A literal shall consist of a single character enclosed in single-quotes (`' '`). All of the escape  
 42275      sequences supported for character constants by the ISO C standard shall be supported by `yacc`.

42276      The relationship with the lexical analyzer is discussed in detail below.

42277      The application shall ensure that the NUL character is not used in grammar rules or literals.

42278      **Declarations Section**

42279      The declarations section is used to define the symbols used to define the target language and  
 42280      their relationship with each other. In particular, much of the additional information required to  
 42281      resolve ambiguities in the context-free grammar for the target language is provided here.

42282      Usually `yacc` assigns the relationship between the symbolic names it generates and their  
 42283      underlying numeric value. The declarations section makes it possible to control the assignment  
 42284      of these values.

42285      It is also possible to keep semantic information associated with the tokens currently on the parse  
 42286      stack in a user-defined C-language **union**, if the members of the union are associated with the  
 42287      various names in the grammar. The declarations section provides for this as well.

42288      The first group of declarators below all take a list of names as arguments. That list can optionally  
 42289      be preceded by the name of a C union member (called a *tag* below) appearing within `'<'` and  
 42290      `'>'`. (As an exception to the typographical conventions of the rest of this volume of  
 42291      IEEE Std. 1003.1-200x, in this case `<tag>` does not represent a metavariable, but the literal angle  
 42292      bracket characters surrounding a symbol.) The use of *tag* specifies that the tokens named on this  
 42293      line shall be of the same C type as the union member referenced by *tag*. This is discussed in  
 42294      more detail below.

42295      For lists used to define tokens, the first appearance of a given token can be followed by a  
 42296      positive integer (as a string of decimal digits). If this is done, the underlying value assigned to it  
 42297      for lexical purposes is taken to be that number.

42298      `%token [<tag>] name [number][name [number]]. . .`

42299      Declares *names* to be a token. If *tag* is present, the C type for all tokens on this line shall be  
 42300      declared to be the type referenced by *tag*. If a positive integer, *number*, follows a *name*, that  
 42301      value shall be assigned to the token.

42302      `%left [<tag>] name [number][name [number]]. . .`

42303      `%right [<tag>] name [number][name [number]]. . .`

42304      Declares *name* to be a token, and assigns precedence to it. One or more lines, each beginning  
 42305      with one of these symbols, can appear in this section. All tokens on the same line have the  
 42306      same precedence level and associativity; the lines are in order of increasing precedence or  
 42307      binding strength. `%left` denotes that the operators on that line are left associative, and  
 42308      `%right` similarly denotes right associative operators. If *tag* is present, it shall declare a C  
 42309      type for *names* as described for `%token`.

42310       %nonassoc [<tag>] *name* [*number*][*name* [*number*]]. . .  
 42311           Declares *name* to be a token, and indicates that this cannot be used associatively. If the  
 42312           parser encounters associative use of this token it reports an error. If *tag* is present, it shall  
 42313           declare a C type for *names* as described for **%token**.

42314       %type [<tag>] *name*. . .  
 42315           Declares that union member *names* are non-terminals, and thus it is required to have a *tag*  
 42316           field at its beginning. Because it deals with non-terminals only, assigning a token number or  
 42317           using a literal is also prohibited. If this construct is present, *yacc* shall perform type  
 42318           checking; if this construct is not present, the parse stack shall hold only the **int** type.

42319       Every name used in *grammar* undefined by a **%token**, **%left**, **%right**, or **%nonassoc** declaration is  
 42320       assumed to represent a non-terminal symbol. The *yacc* utility shall report an error for any non-  
 42321       terminal symbol that does not appear on the left side of at least one grammar rule.

42322       Once the type, precedence, or token number of a name is specified, it shall not be changed. If the  
 42323       first declaration of a token does not assign a token number, *yacc* shall assign a token number.  
 42324       Once this assignment is made, the token number shall not be changed by explicit assignment.

42325       The following declarators do not follow the previous pattern.

42326       %start *name*  
 42327           Declares the non-terminal *name* to be the *start symbol*, which represents the largest, most  
 42328           general structure described by the grammar rules. By default, it is the left-hand side of the  
 42329           first grammar rule; this default can be overridden with this declaration.

42330       %union { *body of union (in C)* }  
 42331           Declares the *yacc* value stack to be a union of the various types of values desired. By default,  
 42332           the values returned by actions (see below) and the lexical analyzer shall be integers. The  
 42333           *yacc* utility keeps track of types, and it shall insert corresponding union member names in  
 42334           order to perform strict type checking of the resulting parser.

42335           Alternatively, given that at least one <tag> construct is used, the union can be declared in a  
 42336           header file (which shall be included in the declarations section by using an **#include**  
 42337           construct within %{ and %}), and a **typedef** used to define the symbol YYSTYPE to  
 42338           represent this union. The effect of **%union** is to provide the declaration of YYSTYPE directly  
 42339           from the *yacc* input.

42340       %{ ... %}  
 42341           C-language declarations and definitions can appear in the declarations section, enclosed by  
 42342           these marks. These statements shall be copied into the code file, and have global scope  
 42343           within it so that they can be used in the rules and program sections.

42344       The application shall ensure that the declarations section is terminated by the token %%. |

#### 42345       **Grammar Rules in yacc**

42346       The rules section defines the context-free grammar to be accepted by the function *yacc* generates,  
 42347       and associates with those rules C-language actions and additional precedence information. The  
 42348       grammar is described below, and a formal definition follows.

42349       The rules section is comprised of one or more grammar rules. A grammar rule has the form:

42350       A : BODY ;

42351       The symbol **A** represents a non-terminal name, and **BODY** represents a sequence of zero or  
 42352       more *names*, *literals*, and *semantic actions* that can then be followed by optional *precedence rules*.  
 42353       Only the names and literals participate in the formation of the grammar; the semantic actions  
 42354       and precedence rules are used in other ways. The colon and the semicolon are *yacc* punctuation.

42355 If there are several successive grammar rules with the same left-hand side, the vertical bar ' | '   
 42356 can be used to avoid rewriting the left-hand side; in this case the semicolon appears only after   
 42357 the last rule. The BODY part can be empty (or empty of names and literals) to indicate that the   
 42358 non-terminal symbol matches the empty string.

42359 The yacc utility assigns a unique number to each rule. Rules using the vertical bar notation are   
 42360 distinct rules. The number assigned to the rule appears in the description file.

42361 The elements comprising a BODY are:

42362 *name, literal*

42363 These form the rules of the grammar: *name* is either a *token* or a *non-terminal*; *literal*   
 42364 stands for itself (less the lexically required quotation marks).

42365 *semantic action*

42366 With each grammar rule, the user can associate actions to be performed each time   
 42367 the rule is recognized in the input process. (Note that the word "action" can also   
 42368 refer to the actions of the parser—shift, reduce, and so on.)

42369 These actions can return values and can obtain the values returned by previous   
 42370 actions. These values are kept in objects of type YYSTYPE (see %union). The   
 42371 result value of the action shall be kept on the parse stack with the left-hand side of   
 42372 the rule, to be accessed by other reductions as part of their right-hand side. By   
 42373 using the <tag> information provided in the declarations section, the code   
 42374 generated by yacc can be strictly type checked and contain arbitrary information. In   
 42375 addition, the lexical analyzer can provide the same kinds of values for tokens, if   
 42376 desired.

42377 An action is an arbitrary C statement and as such can do input or output, call   
 42378 subprograms and alter external variables. An action is one or more C statements   
 42379 enclosed in curly braces ' { ' and ' } '.

42380 Certain pseudo-variables can be used in the action. These are macros for access to   
 42381 data structures known internally to yacc.

42382 \$\$ The value of the action can be set by assigning it to \$\$ . If type   
 42383 checking is enabled and the type of the value to be assigned cannot   
 42384 be determined, a diagnostic message may be generated.

42385 \$number This refers to the value returned by the component specified by the   
 42386 token *number* in the right side of a rule, reading from left to right;   
 42387 *number* can be zero or negative. If it is, it refers to the data associated   
 42388 with the name on the parser's stack preceding the leftmost symbol of   
 42389 the current rule. (That is, "\$0" refers to the name immediately   
 42390 preceding the leftmost name in the current rule, to be found on the   
 42391 parser's stack and "\$-1" refers to the symbol to *its* left.) If *number*   
 42392 refers to an element past the current point in the rule, or beyond the   
 42393 bottom of the stack, the result is undefined. If type checking is   
 42394 enabled and the type of the value to be assigned cannot be   
 42395 determined, a diagnostic message may be generated.

42396 \$<tag>number These correspond exactly to the corresponding symbols without the   
 42397 *tag* inclusion, but allow for strict type checking (and preclude   
 42398 unwanted type conversions). The effect is that the macro is expanded   
 42399 to use *tag* to select an element from the YYSTYPE union (using   
 42400 *dataname.tag*). This is particularly useful if *number* is not positive.   
 42401

|       |                                    |                                                                                                                     |
|-------|------------------------------------|---------------------------------------------------------------------------------------------------------------------|
| 42402 | <code>\$&lt;tag&gt;\$</code>       | This imposes on the reference the type of the union member                                                          |
| 42403 |                                    | referenced by <i>tag</i> . This construction is applicable when a reference                                         |
| 42404 |                                    | to a left context value occurs in the grammar, and provides yacc with                                               |
| 42405 |                                    | a means for selecting a type.                                                                                       |
| 42406 |                                    | Actions can occur in the middle of a rule as well as at the end; an action can access                               |
| 42407 |                                    | values returned by actions to its left, and in turn the value it returns can be                                     |
| 42408 |                                    | accessed by actions to its right. An action appearing in the middle of a rule shall be                              |
| 42409 |                                    | equivalent to replacing the action with a new non-terminal symbol and adding an                                     |
| 42410 |                                    | empty rule with that non-terminal symbol on the left-hand side. The semantic                                        |
| 42411 |                                    | action associated with the new rule shall be equivalent to the original action. The                                 |
| 42412 |                                    | use of actions within rules might introduce conflicts that would not otherwise                                      |
| 42413 |                                    | exist.                                                                                                              |
| 42414 |                                    | By default, the value of a rule shall be the value of the first element in it. If the first                         |
| 42415 |                                    | element does not have a type (particularly in the case of a literal) and type                                       |
| 42416 |                                    | checking is turned on by <code>%type</code> an error message shall result.                                          |
| 42417 | <i>precedence</i>                  | The keyword <code>%prec</code> can be used to change the precedence level associated with a                         |
| 42418 |                                    | particular grammar rule. Examples of this are in cases where a unary and binary                                     |
| 42419 |                                    | operator have the same symbolic representation, but need to be given different                                      |
| 42420 |                                    | precedences, or where the handling of an ambiguous if-else construction is                                          |
| 42421 |                                    | necessary. The reserved symbol <code>%prec</code> can appear immediately after the body of                          |
| 42422 |                                    | the grammar rule and can be followed by a token name or a literal. It shall cause                                   |
| 42423 |                                    | the precedence of the grammar rule to become that of the following token name or                                    |
| 42424 |                                    | literal. The action for the rule as a whole can follow <code>%prec</code> .                                         |
| 42425 |                                    | If a program section follows, the application shall ensure that the grammar rules are terminated                    |
| 42426 |                                    | by <code>%%</code> .                                                                                                |
| 42427 | <b>Programs Section</b>            |                                                                                                                     |
| 42428 |                                    | The <i>programs</i> section can include the definition of the lexical analyzer <code>yylex()</code> , and any other |
| 42429 |                                    | functions, for example those used in the actions specified in the grammar rules. It is unspecified                  |
| 42430 |                                    | whether the programs section precedes or follows the semantic actions in the output file;                           |
| 42431 |                                    | therefore, if the application contains any macro definitions and declarations intended to apply to                  |
| 42432 |                                    | the code in the semantic actions, it shall place them within " <code>%{ . . . %}</code> " in the declarations       |
| 42433 |                                    | section.                                                                                                            |
| 42434 | <b>Input Grammar</b>               |                                                                                                                     |
| 42435 |                                    | The following input to yacc yields a parser for the input to yacc. This formal syntax takes                         |
| 42436 |                                    | precedence over the preceding text syntax description.                                                              |
| 42437 |                                    | The lexical structure is defined less precisely; <b>Lexical Structure of the Grammar</b> on page 1111               |
| 42438 |                                    | defines most terms. The correspondence between the previous terms and the tokens below is as                        |
| 42439 |                                    | follows.                                                                                                            |
| 42440 | <b>IDENTIFIER</b>                  | This corresponds to the concept of <i>name</i> , given previously. It also includes                                 |
| 42441 |                                    | literals as defined previously.                                                                                     |
| 42442 | <b>C_IDENTIFIER</b>                | This is a name, and additionally it is known to be followed by a colon. A literal                                   |
| 42443 |                                    | cannot yield this token.                                                                                            |
| 42444 | <b>NUMBER</b>                      | A string of digits (a non-negative decimal integer).                                                                |
| 42445 | <b>TYPE, LEFT, MARK, and so on</b> |                                                                                                                     |
| 42446 |                                    | These correspond directly to <code>%type</code> , <code>%left</code> , <code>%%</code> , and so on.                 |

42447            {...}            This indicates C-language source code, with the possible inclusion of '\$'  
 42448                               macros as discussed previously.

### 42449 **Notes to Reviewers**

42450            *This section with side shading will not appear in the final copy. - Ed.*

42451            D1, XCU, ERN 375 says that }% should be replaced by %} for RCURL below, but the text is as per  
 42452            existing .2. Comments?

```

42453 /* Grammar for the input to yacc. */
42454 /* Basic entries. */
42455 /* The following are recognized by the lexical analyzer. */

42456 %token IDENTIFIER /* Includes identifiers and literals */
42457 %token C_IDENTIFIER /* identifier (but not literal)
42458 followed by a :. */
42459 %token NUMBER /* [0-9][0-9]* */

42460 /* Reserved words : %type=>TYPE %left=>LEFT, and so on */

42461 %token LEFT RIGHT NONASSOC TOKEN PREC TYPE START UNION

42462 %token MARK /* The %% mark. */
42463 %token LCURL /* The %{ mark. */
42464 %token RCURL /* The }% mark. */

42465 /* 8-bit character literals stand for themselves; */
42466 /* tokens have to be defined for multi-byte characters. */

42467 %start spec
42468 %%

42469 spec : defs MARK rules tail
42470 ;
42471 tail : MARK
42472 {
42473 /* In this action, set up the rest of the file. */
42474 }
42475 /* Empty; the second MARK is optional. */
42476 ;
42477 defs : /* Empty. */
42478 | defs def
42479 ;
42480 def : START IDENTIFIER
42481 | UNION
42482 {
42483 /* Copy union definition to output. */
42484 }
42485 | LCURL
42486 {
42487 /* Copy C code to output file. */
42488 }
42489 | RCURL
42490 | rword tag nlist
42491 ;
42492 rword : TOKEN

```

```

42493 | LEFT
42494 | RIGHT
42495 | NONASSOC
42496 | TYPE
42497 ;
42498 tag : /* Empty: union tag ID optional. */
42499 | '<' IDENTIFIER '>'
42500 ;
42501 nlist : nmno
42502 | nlist nmno
42503 ;
42504 nmno : IDENTIFIER /* Note: literal invalid with % type. */
42505 | IDENTIFIER NUMBER /* Note: invalid with % type. */
42506 ;

42507 /* Rule section */

42508 rules : C_IDENTIFIER rbody prec
42509 | rules rule
42510 ;
42511 rule : C_IDENTIFIER rbody prec
42512 | '|' rbody prec
42513 ;
42514 rbody : /* empty */
42515 | rbody IDENTIFIER
42516 | rbody act
42517 ;
42518 act : '{'
42519 {
42520 /* Copy action, translate $$, and so on. */
42521 }
42522 '}'
42523 ;
42524 prec : /* Empty */
42525 | PREC IDENTIFIER
42526 | PREC IDENTIFIER act
42527 | prec ';'
42528 ;

```

## 42529 Conflicts

42530 The parser produced for an input grammar may contain states in which conflicts occur. The  
 42531 conflicts occur because the grammar is not LALR(1). An ambiguous grammar always contains at  
 42532 least one LALR(1) conflict. The yacc utility shall resolve all conflicts, using either default rules or  
 42533 user-specified precedence rules.

42534 Conflicts are either shift/reduce conflicts or reduce/reduce conflicts. A shift/reduce conflict is  
 42535 where, for a given state and lookahead symbol, both a shift action and a reduce action are  
 42536 possible. A reduce/reduce conflict is where, for a given state and lookahead symbol, reductions  
 42537 by two different rules are possible.

42538 The rules below describe how to specify what actions to take when a conflict occurs. Not all  
 42539 shift/reduce conflicts can be successfully resolved this way because the conflict may be due to  
 42540 something other than ambiguity, so incautious use of these facilities can cause the language



accepted by the parser to be much different from that which was intended. The description file shall contain sufficient information to understand the cause of the conflict. Where ambiguity is the reason either the default or explicit rules should be adequate to produce a working parser.

The declared precedences and associativities (see **Declarations Section** on page 1111) are used to resolve parsing conflicts as follows:

1. A precedence and associativity is associated with each grammar rule; it is the precedence and associativity of the last token or literal in the body of the rule. If the **%prec** keyword is used, it overrides this default. Some grammar rules might not have both precedence and associativity.
2. If there is a shift/reduce conflict, and both the grammar rule and the input symbol have precedence and associativity associated with them, then the conflict is resolved in favor of the action (shift or reduce) associated with the higher precedence. If the precedences are the same, then the associativity is used; left associative implies reduce, right associative implies shift, and non-associative implies an error in the string being parsed.
3. When there is a shift/reduce conflict that cannot be resolved by rule 2, the shift is done. Conflicts resolved this way are counted in the diagnostic output described in **Error Handling**.
4. When there is a reduce/reduce conflict, a reduction is done by the grammar rule that occurs earlier in the input sequence. Conflicts resolved this way are counted in the diagnostic output described in **Error Handling**.

Conflicts resolved by precedence or associativity shall not be counted in the shift/reduce and reduce/reduce conflicts reported by yacc on either standard error or in the description file.

## **Error Handling**

The token **error** shall be reserved for error handling. The name **error** can be used in grammar rules. It indicates places where the parser can recover from a syntax error. The default value of **error** shall be 256. Its value can be changed using a **%token** declaration. The lexical analyzer should not return the value of **error**. (Multi-byte characters should be recognized by the lexical analyzer and returned as tokens. They should not be returned as multi-byte character literals. The token **error** that is used for error recovery is normally assigned the value 256 in the historical implementation. Thus, the token value 256, which used in many multi-byte character sets, is not available for use as the value of a user-defined token.)

The parser shall detect a syntax error when it is in a state where the action associated with the lookahead symbol is **error**. A semantic action can cause the parser to initiate error handling by executing the macro **YYERROR**. When **YYERROR** is executed, the semantic action passes control back to the parser. **YYERROR** cannot be used outside of semantic actions.

When the parser detects a syntax error, it normally calls *yyerror* with the character string "syntax error" as its argument. The call shall not be made if the parser is still recovering from a previous error when the error is detected. The parser is considered to be recovering from a previous error until the parser has shifted over at least three normal input symbols since the last error was detected or a semantic action has executed the macro *yyerrorok*. The parser shall not call *yyerror* when **YYERROR** is executed.

The macro function **YYRECOVERING** shall return 1 if a syntax error has been detected and the parser has not yet fully recovered from it. Otherwise, zero shall be returned.

When a syntax error is detected by the parser, the parser shall check if a previous syntax error has been detected. If a previous error was detected, and if no normal input symbols have been shifted since the preceding error was detected, the parser checks if the lookahead symbol is an

42587 endmarker (see **Interface to the Lexical Analyzer**). If it is, the parser shall return with a non-  
42588 zero value. Otherwise, the lookahead symbol shall be discarded and normal parsing shall  
42589 resume.

42590 When YYERROR is executed or when the parser detects a syntax error and no previous error has  
42591 been detected, or at least one normal input symbol has been shifted since the previous error was  
42592 detected, the parser shall pop back one state at a time until the parse stack is empty or the  
42593 current state allows a shift over **error**. If the parser empties the parse stack, it shall return with a  
42594 non-zero value. Otherwise, it shall shift over **error** and then resume normal parsing. If the parser  
42595 reads a lookahead symbol before the error was detected, that symbol shall still be the lookahead  
42596 symbol when parsing is resumed.

42597 The macro *yterror* in a semantic action shall cause the parser to act as if it has fully recovered  
42598 from any previous errors. The macro *yyclearin* shall cause the parser to discard the current  
42599 lookahead token. If the current lookahead token has not yet been read, *yyclearin* shall have no  
42600 effect.

42601 The macro YYACCEPT shall cause the parser to return with the value zero. The macro  
42602 YYABORT shall cause the parser to return with a non-zero value.

### 42603 **Interface to the Lexical Analyzer**

42604 The *yylex* function is an integer-valued function that returns a *token number* representing the kind  
42605 of token read. If there is a value associated with the token returned by *yylex* (see the discussion  
42606 of *tag* above), it shall be assigned to the external variable *yylval*.

42607 If the parser and *yylex* do not agree on these token numbers, reliable communication between  
42608 them cannot occur. For (one character) literals, the token is simply the numeric value of the  
42609 character in the current character set. The numbers for other tokens can either be chosen by *yacc*,  
42610 or chosen by the user. In either case, the **#define** construct of C is used to allow *yylex* to return  
42611 these numbers symbolically. The **#define** statements are put into the code file, and the header  
42612 file if that file is requested. The set of characters permitted by *yacc* in an identifier is larger than  
42613 that permitted by C. Token names found to contain such characters shall not be included in the  
42614 **#define** declarations.

42615 If the token numbers are chosen by *yacc*, the tokens other than literals shall be assigned numbers  
42616 greater than 256, although no order is implied. A token can be explicitly assigned a number by  
42617 following its first appearance in the declarations section with a number. Names and literals not  
42618 defined this way retain their default definition. All token numbers assigned by *yacc* shall be  
42619 unique and distinct from the token numbers used for literals and user-assigned tokens. If  
42620 duplicate token numbers cause conflicts in parser generation, *yacc* shall report an error;  
42621 otherwise, it is unspecified whether the token assignment is accepted or an error is reported.

42622 The end of the input is marked by a special token called the *endmarker*, which has a token  
42623 number that is zero or negative. (These values are invalid for any other token.) All lexical  
42624 analyzers shall return zero or negative as a token number upon reaching the end of their input. If  
42625 the tokens up to, but excluding, the endmarker form a structure that matches the start symbol,  
42626 the parser shall accept the input. If the endmarker is seen in any other context, it shall be  
42627 considered an error.

## 42628 **Completing the Program**

42629 In addition to *yyparse* and *yylex*, the functions *yyerror* and *main* are required to make a complete  
 42630 program. The application can supply *main* and *yyerror*, or those routines can be obtained from  
 42631 the *yacc* library.

## 42632 **Yacc Library**

42633 The following functions appear only in the *yacc* library accessible through the `-ly` operand to *cc*  
 42634 or *c89*; they can therefore be redefined by a portable application:

### 42635 **int main(void)**

42636 This function shall call *yyparse* and exit with an unspecified value. Other actions within this  
 42637 function are unspecified.

### 42638 **int yyerror(const char \*s)**

42639 This function shall write the NUL-terminated argument to standard error, followed by a  
 42640 <newline> character.

42641 The order of the `-ly` and `-ll` operands given to *cc* or *c89* is significant; the application shall  
 42642 either provide its own *main* function or ensure that `-ly` precedes `-ll`.

## 42643 **Debugging the Parser**

42644 The parser generated by *yacc* shall have diagnostic facilities in it that can be optionally enabled  
 42645 at either compile time or at runtime (if enabled at compile time). The compilation of the runtime  
 42646 debugging code is under the control of *YYDEBUG*, a preprocessor symbol. If *YYDEBUG* has a  
 42647 non-zero value, the debugging code shall be included. If its value is zero, the code shall not be  
 42648 included.

42649 In parsers where the debugging code has been included, the external **int** *yydebug* can be used to  
 42650 turn debugging on (with a non-zero value) and off (zero value) at runtime. The initial value of  
 42651 *yydebug* shall be zero.

42652 When `-t` is specified, the code file shall be built such that, if *YYDEBUG* is not already defined at  
 42653 compilation time (using the *c89* `-D YYDEBUG` option, for example), *YYDEBUG* shall be set  
 42654 explicitly to 1. When `-t` is not specified, the code file shall be built such that, if *YYDEBUG* is not  
 42655 already defined, it shall be set explicitly to zero.

42656 The format of the debugging output is unspecified but includes at least enough information to  
 42657 determine the shift and reduce actions, and the input symbols. It also provides information  
 42658 about error recovery.

## 42659 **Algorithms**

42660 The parser constructed by *yacc* implements an LALR(1) parsing algorithm as documented in the  
 42661 literature. It is unspecified whether the parser is table-driven or direct-coded.

42662 A parser generated by *yacc* shall never request an input symbol from *yylex* while in a state where  
 42663 the only actions other than the error action are reductions by a single rule.

42664 The literature of parsing theory defines these concepts.

**Limits**

The yacc utility may have several internal tables. The minimum maximums for these tables are shown in the following table. The exact meaning of these values is implementation-dependent. The implementation shall define the relationship between these values and between them and any error messages that the implementation may generate should it run out of space for any internal structure. An implementation may combine groups of these resources into a single pool as long as the total available to the user does not fall below the sum of the sizes specified by this section.

**Table 4-22** Internal Limits in yacc

| Limit      | Minimum<br>Maximum | Description                                                                                                                                                                                                                                             |
|------------|--------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| {NTERMS}   | 126                | Number of tokens.                                                                                                                                                                                                                                       |
| {NNONTERM} | 200                | Number of non-terminals.                                                                                                                                                                                                                                |
| {NPROD}    | 300                | Number of rules.                                                                                                                                                                                                                                        |
| {NSTATES}  | 600                | Number of states.                                                                                                                                                                                                                                       |
| {MEMSIZE}  | 5 200              | Length of rules. The total length, in names (tokens and non-terminals), of all the rules of the grammar. The left-hand side is counted for each rule, even if it is not explicitly repeated, as specified in <b>Grammar Rules in yacc</b> on page 1112. |
| {ACTSIZE}  | 4 000              | Number of actions. “Actions” here (and in the description file) refer to parser actions (shift, reduce, and so on) not to semantic actions defined in <b>Grammar Rules in yacc</b> on page 1112.                                                        |

**EXIT STATUS**

The following exit values shall be returned:

0 Successful completion.

>0 An error occurred.

**CONSEQUENCES OF ERRORS**

If any errors are encountered, the run is aborted and yacc exits with a non-zero status. Partial code files and header files may be produced. The summary information in the description file always shall be produced if the **-v** flag is present.

**APPLICATION USAGE**

Historical implementations experience name conflicts on the names **yacc.tmp**, **yacc.acts**, **yacc.debug**, **y.tab.c**, **y.tab.h**, and **y.output** if more than one copy of yacc is running in a single directory at one time. The **-b** option was added to overcome this problem. The related problem of allowing multiple yacc parsers to be placed in the same file was addressed by adding a **-p** option to override the previously hard-coded yy variable prefix.

The description of the **-p** option specifies the minimal set of function and variable names that cause conflict when multiple parsers are linked together. YYSTYPE does not need to be changed. Instead, the programmer can use **-b** to give the header files for different parsers different names, and then the file with the yylex for a given parser can include the header for that parser. Names such as **yyclearerr** do not need to be changed because they are used only in the actions; they do not have linkage. It is possible that an implementation has other names, either internal ones for implementing things such as **yyclearerr**, or providing non-standard features that it wants to change with **-p**.

42712 Unary operators that are the same token as a binary operator in general need their precedence  
 42713 adjusted. This is handled by the **%prec** advisory symbol associated with the particular grammar  
 42714 rule defining that unary operator. (See **Grammar Rules in yacc** on page 1112.) Applications are  
 42715 not required to use this operator for unary operators, but the grammars that do not require it are  
 42716 rare.

#### 42717 EXAMPLES

42718 Access to the *yacc* library is obtained with library search operands to *cc* or *c89*. To use the *yacc*  
 42719 library *main*:

```
42720 c89 y.tab.c -l y
```

42721 Both the *lex* library and the *yacc* library contain *main*. To access the *yacc main*:

```
42722 c89 y.tab.c lex.yy.c -l y -l l
```

42723 This ensures that the *yacc* library is searched first, so that its *main* is used.

42724 The historical *yacc* libraries have contained two simple functions that are normally coded by the  
 42725 application programmer. These library functions are similar to the following code:

```
42726 #include <locale.h>
42727 int main(void)
42728 {
42729 extern int yyparse();
42730 setlocale(LC_ALL, "");
42731 /* If the following parser is one created by lex, the
42732 application must be careful to ensure that LC_CTYPE
42733 and LC_COLLATE are set to the POSIX locale. */
42734 (void) yyparse();
42735 return (0);
42736 }
42737 #include <stdio.h>
42738 int yyerror(const char *msg)
42739 {
42740 (void) fprintf(stderr, "%s\n", msg);
42741 return (0);
42742 }
```

#### 42743 RATIONALE

42744 The references in **Referenced Documents** on page xv may be helpful in constructing the parser  
 42745 generator. The referenced DeRemer, Frank, and Pennello Article (along with the works it  
 42746 references) describes a technique to generate parsers that conform to this volume of  
 42747 IEEE Std. 1003.1-200x. Work in this area continues to be done, so implementors should consult  
 42748 current literature before doing any new implementations. The original Knuth Article is the  
 42749 theoretical basis for this kind of parser, but the tables it generates are impractically large for  
 42750 reasonable grammars and should not be used. The “equivalent to” wording is intentional to  
 42751 assure that the best tables that are LALR(1) can be generated.

42752 There has been confusion between the class of grammars, the algorithms needed to generate  
 42753 parsers, and the algorithms needed to parse the languages. They are all reasonably orthogonal.  
 42754 In particular, a parser generator that accepts the full range of LR(1) grammars need not generate  
 42755 a table any more complex than one that accepts SLR(1) (a relatively weak class of LR grammars)  
 42756 for a grammar that happens to be SLR(1). Such an implementation need not recognize the case,  
 42757 either; table compression can yield the SLR(1) table (or one even smaller than that) without

recognizing that the grammar is SLR(1). The speed of an LR(1) parser for any class is dependent more upon the table representation and compression (or the code generation if a direct parser is generated) than upon the class of grammar that the table generator handles.

The speed of the parser generator is somewhat dependent upon the class of grammar it handles. However, the original Knuth Article algorithms for constructing LR parsers was judged by its author to be impractically slow at that time. Although full LR is more complex than LALR(1), as computer speeds and algorithms improve, the difference (in terms of acceptable wall-clock execution time) is becoming less significant.

Potential authors are cautioned that the referenced DeRemer, Frank, and Pennello Article previously cited identifies a bug (an over-simplification of the computation of LALR(1) lookahead sets) in some of the LALR(1) algorithm statements that preceded it to publication. They should take the time to seek out that paper, as well as current relevant work, particularly Aho's.

The **-b** option was added to provide a portable method for permitting yacc to work on multiple separate parsers in the same directory. If a directory contains more than one yacc grammar, and both grammars are constructed at the same time (by, for example, a parallel *make* program), conflict results. While the solution is not historical practice, it corrects a known deficiency in historical implementations. Corresponding changes were made to all sections that referenced the file names **y.tab.c** (now "the code file"), **y.tab.h** (now "the header file"), and **y.output** (now "the description file").

The grammar for yacc input is based on System V documentation. The textual description shows there that the ' ; ' is required at the end of the rule. The grammar and the implementation do not require this. (The use of **C\_IDENTIFIER** causes a reduce to occur in the right place.)

Also, in that implementation, the constructs such as **%token** can be terminated by a semicolon, but this is not permitted by the grammar. The keywords such as **%token** can also appear in uppercase, which is again not discussed. In most places where '%' is used, '\' can be substituted, and there are alternate spellings for some of the symbols (for example, **%LEFT** can be "%<" or even "\<").

Historically, **<tag>** can contain any characters except '>', including white space, in the implementation. However, since the *tag* must reference a ISO C standard union member, in practice conforming implementations need to support only the set of characters for ISO C standard identifiers in this context.

Some historical implementations are known to accept actions that are terminated by a period. Historical implementations often allow '\$' in names. A conforming implementation does not need to support either of these behaviors.

Deciding when to use **%prec** illustrates the difficulty in specifying the behavior of yacc. There may be situations in which the *grammar* is not, strictly speaking, in error, and yet yacc cannot interpret it unambiguously. The resolution of ambiguities in the grammar can in many instances be resolved by providing additional information, such as using **%type** or **%union** declarations. It is often easier and it usually yields a smaller parser to take this alternative when it is appropriate.

The size and execution time of a program produced without the runtime debugging code is usually smaller and slightly faster in historical implementations.

Statistics messages from several historical implementations include the following types of information:

|       |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|-------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 42803 | <i>n</i> /512 terminals, <i>n</i> /300 non-terminals                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| 42804 | <i>n</i> /600 grammar rules, <i>n</i> /1 500 states                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 42805 | <i>n</i> shift/reduce, <i>n</i> reduce/reduce conflicts reported                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 42806 | <i>n</i> /350 working sets used                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 42807 | Memory: states, etc. <i>n</i> /15 000, parser <i>n</i> /15 000                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 42808 | <i>n</i> /600 distinct lookahead sets                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 42809 | <i>n</i> extra closures                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 42810 | <i>n</i> shift entries, <i>n</i> exceptions                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| 42811 | <i>n</i> goto entries                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 42812 | <i>n</i> entries saved by goto default                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| 42813 | Optimizer space used: input <i>n</i> /15 000, output <i>n</i> /15 000                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 42814 | <i>n</i> table entries, <i>n</i> zero                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 42815 | Maximum spread: <i>n</i> , Maximum offset: <i>n</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 42816 | The report of internal tables in the description file is left implementation-dependent because all aspects of these limits are also implementation-dependent. Some implementations may use dynamic allocation techniques and have no specific limit values to report.                                                                                                                                                                                                                                                                     |
| 42817 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 42818 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 42819 | The format of the <b>y.output</b> file is not given because specification of the format was not seen to enhance application portability. The listing is primarily intended to help human users understand and debug the parser; use of <b>y.output</b> by a portable application script would be unusual. Furthermore, implementations have not produced consistent output and no popular format was apparent. The format selected by the implementation should be human-readable, in addition to the requirement that it be a text file. |
| 42820 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 42821 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 42822 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 42823 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 42824 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 42825 | Standard error reports are not specifically described because they are seldom of use to portable applications and there was no reason to restrict implementations.                                                                                                                                                                                                                                                                                                                                                                        |
| 42826 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 42827 | Some implementations recognize " <code>= {</code> " as equivalent to " <code>' {'</code> " because it appears in historical documentation. This construction was recognized and documented as obsolete as long ago as 1978, in the referenced <i>Yacc: Yet Another Compiler-Compiler</i> . This volume of IEEE Std. 1003.1-200x chose to leave it as obsolete and omit it.                                                                                                                                                                |
| 42828 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 42829 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 42830 |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 42831 | <b>FUTURE DIRECTIONS</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| 42832 | None.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 42833 | <b>SEE ALSO</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 42834 | <i>c89</i> , <i>lex</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| 42835 | <b>CHANGE HISTORY</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| 42836 | First released in Issue 2.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| 42837 | <b>Issue 4</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 42838 | Aligned with the ISO/IEC 9945-2:1993 standard.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 42839 | <b>Issue 5</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 42840 | FUTURE DIRECTIONS section added.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| 42841 | <b>Issue 6</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| 42842 | Minor changes have been added to align with the IEEE P1003.2b draft standard.                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 42843 | The normative text is reworded to avoid use of the term “must” for application requirements.                                                                                                                                                                                                                                                                                                                                                                                                                                              |

42844 **NAME**

42845           zcat — expand and concatenate data

42846 **SYNOPSIS**42847 XSI       zcat [*file...*]

42848

42849 **DESCRIPTION**

42850       The *zcat* utility shall write to standard output the uncompressed form of files that have been  
 42851       compressed using the *compress* utility. It is the equivalent of *uncompress -c*. Input files are not  
 42852       affected.

42853 **OPTIONS**

42854       None.

42855 **OPERANDS**

42856       The following operand shall be supported:

42857       *file*           The path name of a file previously processed by the *compress* utility. If *file* already  
 42858       has the *.Z* suffix specified, it is used as submitted. Otherwise, the *.Z* suffix is  
 42859       appended to the file name prior to processing.

42860 **STDIN**42861       The standard input shall be used only if no *file* operands are specified, or if a *file* operand is *'-'*.42862 **INPUT FILES**42863       Input files shall be compressed files that are in the format produced by the *compress* utility.42864 **ENVIRONMENT VARIABLES**42865       The following environment variables shall affect the execution of *zcat*:

42866       *LANG*           Provide a default value for the internationalization variables that are unset or null.  
 42867       If *LANG* is unset or null, the corresponding value from the implementation-  
 42868       dependent default locale shall be used. If any of the internationalization variables  
 42869       contains an invalid setting, the utility shall behave as if none of the variables had  
 42870       been defined.

42871       *LC\_ALL*          If set to a non-empty string value, override the values of all the other  
 42872       internationalization variables.

42873       *LC\_CTYPE*       Determine the locale for the interpretation of sequences of bytes of text data as  
 42874       characters (for example, single-byte as opposed to multi-byte characters in  
 42875       arguments).

42876       *LC\_MESSAGES*   Determine the locale that should be used to affect the format and contents of  
 42877       diagnostic messages written to standard error.  
 42878

42879       *NLSPATH*       Determine the location of message catalogs for the processing of *LC\_MESSAGES*.

42880 **ASYNCHRONOUS EVENTS**

42881       Default.

42882 **STDOUT**

42883       The compressed files given as input shall be written on standard output in their uncompressed  
 42884       form.



42885 **STDERR**

42886           Used only for diagnostic messages.

42887 **OUTPUT FILES**

42888           None.

42889 **EXTENDED DESCRIPTION**

42890           None.

42891 **EXIT STATUS**

42892           The following exit values shall be returned:

42893           0   Successful completion.

42894           &gt;0  An error occurred.

42895 **CONSEQUENCES OF ERRORS**

42896           Default.

42897 **APPLICATION USAGE**

42898           None.

42899 **EXAMPLES**

42900           None.

42901 **RATIONALE**

42902           None.

42903 **FUTURE DIRECTIONS**

42904           None.

42905 **SEE ALSO**42906           *compress, uncompress*42907 **CHANGE HISTORY**

42908           First released in Issue 4.



## Portability Considerations (Informative)

42910

### Notes to Reviewers

42912 *This section with side shading will not appear in the final copy. - Ed.*

42913 This chapter needs a complete overhaul. Volunteers are wanted to tackle this task.

42914 This chapter contains information to satisfy the recommendations of the TSG-1 Final Report  
42915 Annex A:

42916 Section 5.1 describes perceived user requirements.

42917 • Section 5.2 on page 1129 indicates how the facilities of this volume of IEEE Std. 1003.1-200x  
42918 satisfy those requirements.

42919 • Section 5.3 on page 1132 offers guidance to writers of profiles on how the configurable  
42920 options, limits, and optional behavior of this volume of IEEE Std. 1003.1-200x should be cited  
42921 in profiles.

### 5.1 User Requirements

42923 This section describes the user requirements that were perceived by the developers of this  
42924 volume of IEEE Std. 1003.1-200x. The primary source for these requirements was an analysis of  
42925 historical practice in widespread use, as typified by the base documents listed Chapter 1 on page  
42926 1.

42927 The universe of users applicable to this volume of IEEE Std. 1003.1-200x is a superset of those  
42928 addressed by the System Interfaces volume of IEEE Std. 1003.1-200x: users requiring open  
42929 systems solutions for source-code portability of applications involving multi-programming and  
42930 process management (creating processes, signaling, and so on); access to files and directories in a  
42931 hierarchy of file systems (opening, reading, writing, deleting files, and so on); access to  
42932 asynchronous communications ports and other special devices; access to information about  
42933 other users of the system. The users of the System Interfaces volume of IEEE Std. 1003.1-200x are  
42934 limited to those employing applications written in high-level languages, such as C, Ada, or  
42935 FORTRAN.

42936 The following additional users are identified for this volume of IEEE Std. 1003.1-200x:

- 42937 • Users who desire portable applications that do not necessarily require the characteristics of  
42938 high-level languages (for example, the speed of execution of compiled languages or the  
42939 relative security of source code intellectual property inherent in the compilation process)
- 42940 • Users who desire portable applications that can be developed quickly and can be modified  
42941 readily without the use of compilers and other system components that may be unavailable  
42942 on small systems or those without special application development capabilities
- 42943 • Users who interact with a system to achieve general-purpose time-sharing capabilities  
42944 common to most business or government offices or academic environments: editing, filing,  
42945 inter-user communications, printing, and so on
- 42946 • Users who develop applications for POSIX-conformant systems

42947 An acknowledged restriction on applicable users is that they are limited to the group of  
42948 individuals who are familiar with the style of interaction characteristic of historically-derived

42949 systems based on one of the UNIX operating systems (as opposed to other historical systems  
42950 with different models, such as MS/DOS, Macintosh, VMS, MVS, and so on). Typical users would  
42951 include program developers, engineers, or general-purpose time-sharing users.

42952 The following subsections list the perceived requirements for this universe of users, in addition  
42953 to those identified by the System Interfaces volume of IEEE Std. 1003.1-200x.

#### 42954 **5.1.1 Command Language**

42955 Users should be able to define procedures that combine simple tools and/or applications into  
42956 higher-level components that perform to the specific needs of the user. The user should be able  
42957 to store, recall, use, and modify these procedures. These procedures should employ a powerful  
42958 command language that is used for recurring tasks in portable applications (scripts) in the same  
42959 way that it is used interactively to accomplish one-time tasks. The language and the utilities that  
42960 it uses must be consistent between systems to reduce errors and retraining.

#### 42961 **5.1.2 Interactive Facilities**

42962 Use the system to accomplish individual tasks at an interactive terminal. The interface should be  
42963 consistent, intuitive, and offer usability enhancements to increase the productivity of terminal  
42964 users, reduce errors, and minimize retraining costs. Online documentation or usage assistance  
42965 should be available.

#### 42966 **5.1.3 Accomplish Multiple Tasks Simultaneously**

42967 Access applications and interactive facilities from a single terminal without requiring serial  
42968 execution: switch between multiple interactive tasks; schedule one-time or periodic background  
42969 work; display the status of all work in progress or scheduled; influence the priority scheduling of  
42970 work, when authorized.

#### 42971 **5.1.4 Complex Data Manipulation**

42972 Manipulate data in files in complex ways: sort, merge, compare, translate, edit, format, pattern  
42973 match, select subsets (strings, columns, fields, rows, and so on). These facilities should be  
42974 available to both portable applications and interactive users.

#### 42975 **5.1.5 File Hierarchy Manipulation**

42976 Create, delete, move/rename, copy, backup/archive, and display files and directories. These  
42977 facilities should be available to both portable applications and interactive users.

#### 42978 **5.1.6 Locale Configuration**

42979 Customize applications and interactive sessions for the cultural and language conventions of the  
42980 user. Employ a wide variety of standard character encodings. These facilities should be available  
42981 to both portable applications and interactive users.

**42982 5.1.7 Inter-User Communication**

42983 Send messages or transfer files to other users on the same system or other systems on a network.  
 42984 These facilities should be available to both portable applications and interactive users.

**42985 5.1.8 System Environment**

42986 Display information about the status of the system (activities of users and their interactive and  
 42987 background work, file system utilization, system time, configuration, and presence of optional  
 42988 facilities) and the environment of the user (terminal characteristics, and so on). Inform the  
 42989 system operator/administrator of problems. Control access to user files and other resources.

**42990 5.1.9 Printing**

42991 Output files on a variety of output device classes, accessing devices on local or network-  
 42992 connected systems. Control (or influence) the formatting, priority scheduling, and output  
 42993 distribution of work. These facilities should be available to both portable applications and  
 42994 interactive users.

**42995 5.1.10 Software Development**

42996 Develop (create and manage source files, compile/interpret, debug) portable open systems  
 42997 applications and package them for distribution to, and updating of, other systems.

**42998 5.2 Portability Capabilities**

42999 This section describes the significant portability capabilities of this volume of  
 43000 IEEE Std. 1003.1-200x and indicates how the user requirements listed in Section 5.1 on page 1127  
 43001 are addressed. The capabilities are listed in the same format as the preceding user requirements;  
 43002 they are summarized below:

- 43003 • Command Language
- 43004 • Interactive Facilities
- 43005 • Accomplish Multiple Tasks Simultaneously
- 43006 • Complex Data Manipulation
- 43007 • File Hierarchy Manipulation
- 43008 • Locale Configuration
- 43009 • Inter-user Communication
- 43010 • System Environment
- 43011 • Printing
- 43012 • Software Development

43013 The shell command language, as described in Chapter 2 on page 35, is a common language  
 43014 useful in batch scripts, through an API to high-level languages (for the C-Language Binding  
 43015 option, *system()* and *popen()*) and through an interactive terminal (see *sh* on page 888). The shell  
 43016 language has many of the characteristics of a high-level language, but it has been designed to be  
 43017 more suitable for user terminal entry and includes interactive debugging facilities. Through the  
 43018 use of pipelining, many complex commands can be constructed from combinations of data  
 43019 filters and other common components. Shell scripts can be created, stored, recalled, and

43020 modified by the user with simple editors.

43021 In addition to the basic shell language, the following utilities offer features that simplify and  
 43022 enhance programmatic access to the utilities and provide features normally found only in high-  
 43023 level languages: *basename*, *bc*, *command*, *dirname*, *echo*, *env*, *expr*, *false*, *printf*, *read*, *sleep*, *tee*, *test*,  
 43024 *time*<sup>4</sup>, *true*, *wait*, *xargs*, and all of the special built-in utilities in Section 2.14 on page 96.

#### 43025 **Unsatisfied Requirements**

43026 None.

### 43027 **5.2.1 Interactive Facilities**

43028 The utilities offer a common style of command line interface through conformance to the Utility  
 43029 Syntax Guidelines (see the System Interface Definitions volume of IEEE Std. 1003.1-200x, Section  
 43030 12.2, Utility Syntax Guidelines) and the common utility defaults (see Section 1.11 on page 25).  
 43031 The *sh* utility offers an interactive command line history and editing facility. The following  
 43032 utilities in the User Portability Utilities option have been customized for interactive use: *alias*, *ex*,  
 43033 *fc*, *mailx*, *more*, *talk*, *vi*, *unalias*, and *write*; the *man* utility offers online access to system  
 43034 documentation.

#### 43035 **Unsatisfied Requirements**

43036 The command line interface to individual utilities is as intuitive and consistent as historical  
 43037 practice allows. Work underway based on graphical user interfaces (see Section 1.7 on page 10)  
 43038 may be more suitable for novice or occasional users of the system.

### 43039 **5.2.2 Accomplish Multiple Tasks Simultaneously**

43040 The shell command language offers background processing through the asynchronous list  
 43041 command form; see Section 2.9.3.1 on page 74. The *nohup* utility makes background processing  
 43042 more robust and usable. The *kill* utility can terminate background jobs. When the User  
 43043 Portability Utilities option and the POSIX.1 Job Control option are supported, the following  
 43044 utilities can support more complex background work: *bg*, *fg*, and *jobs*. With just the User  
 43045 Portability Utilities option, the following can support periodic job scheduling, control, and  
 43046 display: *at*, *batch*, *crontab*, *nice*, *ps*, and *renice*.

#### 43047 **Unsatisfied Requirements**

43048 Terminals with multiple windows may be more suitable for some multi-tasking interactive uses  
 43049 than the job control approach in this volume of IEEE Std. 1003.1-200x. See the comments on  
 43050 graphical user interfaces in Section 5.2.1. The *nice* and *renice* utilities do not necessarily take  
 43051 advantage of complex system scheduling algorithms that are being developed by realtime  
 43052 standards efforts; additional facilities are expected in future versions of this volume of  
 43053 IEEE Std. 1003.1-200x. More sophisticated job processing facilities will be introduced in a future  
 43054 version, based on work in supercomputing standards efforts.

43055 \_\_\_\_\_

43056 4. The utilities listed with an asterisk are present only on systems which support the User Portability Utilities option. There may be  
 43057 further restrictions on the utilities offered with various configuration option combinations; see the individual utility descriptions.

43058 **5.2.3 Complex Data Manipulation**

43059 The following utilities address user requirements in this area: *asa*, *awk*, *bc*, *cmp*, *comm*, *csplit\**, *cut*,  
 43060 *dd*, *diff*, *ed*, *ex\**, *expand\**, *expr*, *find*, *fold*, *grep*, *head*, *join*, *od*, *paste*, *pr*, *printf*, *sed*, *sort*, *split\**, *tabs\**, *tail*,  
 43061 *tr*, *unexpand\**, *uniq*, *uudecode\**, *uencode\**, and *wc*.

43062 **Unsatisfied Requirements**

43063 Sophisticated text formatting utilities, such as *troff* or *TeX*, are not included. Standards work in  
 43064 the area of SGML may satisfy this.

43065 **5.2.4 File Hierarchy Manipulation**

43066 The following utilities address user requirements in this area: *basename*, *cat*, *cd*, *chgrp*, *chmod*,  
 43067 *chown*, *cksum*, *cp*, *dd*, *df\**, *diff*, *dirname*, *du\**, *find*, *ls*, *ln*, *mkdir*, *mkfifo*, *mv*, *patch\**, *pathchk*, *pax*, *pwd*,  
 43068 *rm*, *rmdir*, *test*, and *touch*.

43069 **Unsatisfied Requirements**

43070 Some graphical user interfaces offer more intuitive file manager components that allow file  
 43071 manipulation through the movement of icons for novice users.

43072 **5.2.5 Locale Configuration**

43073 The standard utilities are affected by the various *LC\_* variables to achieve locale-dependent  
 43074 operation: character classification, collation sequences, REs and shell pattern matching,  
 43075 date/time formats, numeric formatting, and monetary formatting. When the  
 43076 POSIX2\_LOCALEDEF option is supported, applications can provide their own locale definition  
 43077 files. The following utilities address user requirements in this area: *date*, *ed*, *ex\**, *find*, *grep*, *locale*,  
 43078 *localedef*, *more\**, *sed*, *sh*, *sort*, *tr*, *uniq*, and *vi\**.

43079 **Unsatisfied Requirements**

43080 Some aspects of multi-byte character and state-encoded character encodings have not yet been  
 43081 addressed. The C-language functions, such as *getopt()*, are generally limited to single-byte  
 43082 characters. The effect of the *LC\_MESSAGES* variable on message formats is only suggested at  
 43083 this time, and utilities for message catalog manipulation have not been defined.

43084 **5.2.6 Inter-User Communication**

43085 The following utilities address user requirements in this area: *cksum*, *mailx\**, *mesg\**, *patch\**, *pax*,  
 43086 *talk\**, *uudecode\**, *uencode\**, *who\**, and *write\**.

43087 **Unsatisfied Requirements**

43088 The historical UUCP utilities are not included. This type of requirement will be addressed as  
 43089 part of networking standards efforts.

43090 **5.2.7 System Environment**

43091 The following utilities address user requirements in this area: *chgrp*, *chmod*, *chown*, *df\**, *du\**, *env*,  
 43092 *getconf*, *id*, *logger*, *logname*, *mesg\**, *newgrp\**, *ps\**, *stty*, *tput\**, *tty*, *umask*, *uname*, and *who\**.

43093 **Unsatisfied Requirements**

43094 Considerable extra control of security, privilege, and auditing facilities will be added in a future  
 43095 version, based on work underway in security standards efforts.

43096 **5.2.8 Printing**

43097 The following utilities address user requirements in this area: *pr* and *lp*.

43098 **Unsatisfied Requirements**

43099 There are no features to control the formatting or scheduling of the print jobs. Such facilities will  
 43100 be added in a future version, based on work underway in system administration standards  
 43101 efforts.

43102 **5.2.9 Software Development**

43103 The following utilities address user requirements in this area: *ar*, *asa*, *awk*, *c89*, *ctags\**, *fort77*,  
 43104 *getconf*, *getopts*, *lex*, *localedef*, *make*, *nm\**, *od*, *patch\**, *pax*, *strings\**, *strip*, *time\**, and *yacc*.

43105 The *system()*, *popen()*, *pclose()*, *regcomp()*, *regexexec()*, *regerror()*, *regfree()*, *fnmatch()*, *getopt()*,  
 43106 *glob()*, *globfree()*, *wordexp()*, and *wordfree()* functions allow C-language programmers to access  
 43107 some of the interfaces used by the utilities, such as argument processing, REs, and pattern  
 43108 matching.

43109 **Unsatisfied Requirements**

43110 There are no language-specific development tools related to languages other than C and  
 43111 FORTRAN. The C tools are more complete and varied than the FORTRAN tools. There is no  
 43112 source-code control system. There is no data dictionary or other CASE-like development tools.

43113 **5.3 Profiling Considerations**

43114 This section offers guidance to writers of profiles on how the configurable options, limits, and  
 43115 optional behavior of this volume of IEEE Std. 1003.1-200x should be cited in profiles. Profile  
 43116 writers should consult the general guidance in POSIX.0 when writing POSIX Standardized  
 43117 Profiles.

43118 The information in this section is an inclusive list of features that should be considered by profile  
 43119 writers. Further subsetting of this volume of IEEE Std. 1003.1-200x, including the specification of  
 43120 behavior currently described as unspecified, undefined, implementation-dependent, or with the  
 43121 verbs “may” or “need not” violates the intent of the developers of this volume of  
 43122 IEEE Std. 1003.1-200x and the guidelines of ISO/IEC TR 10000-1.



### 43123 5.3.1 Configuration Options

43124 There are three broad optional configurations suggested by this volume of IEEE Std. 1003.1-200x:  
 43125 basic execution system, development system, and user portability interactive system. The  
 43126 options to support these, and other minor configuration options, are listed in the System  
 43127 Interface Definitions volume of IEEE Std. 1003.1-200x, Chapter 2, Conformance. Profile writers  
 43128 should consult the following list and the comments concerning user requirements addressed by  
 43129 various POSIX.2 components in Section 5.2 on page 1129.

#### 43130 POSIX2\_UPE

43131 The system supports the User Portability Utilities option.

43132 This option is a requirement for a user portability interactive system. It is required  
 43133 frequently except for those systems, such as embedded realtime or dedicated application  
 43134 systems, that support little or no interactive time-sharing work by users or operators.

#### 43135 POSIX2\_SW\_DEV

43136 The system supports the Software Development Utilities option.

43137 This option is required by many systems, even those in which actual software development  
 43138 does not occur. The *make* utility, in particular, is required by many application software  
 43139 packages as they are installed onto the system. If POSIX2\_C\_DEV is supported,  
 43140 POSIX2\_SW\_DEV is almost a mandatory requirement because of *ar* and *make*.

#### 43141 POSIX2\_C\_BIND

43142 The system supports the C-Language Bindings option.

43143 This option is required on some systems developing complex C applications or on any  
 43144 system installing C applications in source form that require the functions in this option. The  
 43145 *system()* and *popen()* functions, in particular, are widely used by applications; the others are  
 43146 rather more specialized.

#### 43147 POSIX2\_C\_DEV

43148 The system supports the C-Language Development Utilities option.

43149 This option is required by many systems, even those in which actual C-language software  
 43150 development does not occur. The *c89* utility, in particular, is required by many application  
 43151 software packages as they are installed onto the system. The *lex* and *yacc* utilities are used  
 43152 less frequently.

#### 43153 POSIX2\_FORT\_DEV

43154 The system supports the FORTRAN Development Utilities option

43155 As with C, this option is needed on any system developing or installing FORTRAN  
 43156 applications in source form.

#### 43157 POSIX2\_FORT\_RUN

43158 The system supports the FORTRAN Runtime Utilities option.

43159 This option is required for some FORTRAN applications that need the *asa* utility to convert  
 43160 Hollerith printing statement output. It is unknown how frequently this occurs.

#### 43161 POSIX2\_LOCALEDEF

43162 The system supports the creation of locales.

43163 This option is needed if applications require their own customized locale definitions to  
 43164 operate. It is presently unknown whether many applications are dependent on this.  
 43165 However, the option is virtually mandatory for systems in which internationalized  
 43166 applications are developed.

43167 POSIX2\_CHAR\_TERM

43168 The system supports at least one terminal type capable of all operations described in this  
43169 volume of IEEE Std. 1003.1-200x.

43170 On systems with POSIX2\_UPE, this option is almost always required. It was developed  
43171 solely to allow certain specialized vendors and user applications to bypass the requirement  
43172 for general-purpose asynchronous terminal support. For example, an application and  
43173 system that was suitable for block-mode terminals, such as IBM 3270s, would not need this  
43174 option.

### 43175 5.3.2 Configurable Limits

43176 Very few of the limits in Section 1.9 on page 21 need to be increased for profiles. No profile can  
43177 cite lower values.

43178 {POSIX2\_BC\_BASE\_MAX}

43179 {POSIX2\_BC\_DIM\_MAX}

43180 {POSIX2\_BC\_SCALE\_MAX}

43181 {POSIX2\_BC\_STRING\_MAX}

43182 No increase is anticipated for any of these *bc* values, except for very specialized  
43183 applications involving huge numbers.

43184 {POSIX2\_COLL\_WEIGHTS\_MAX}

43185 Some natural languages with complex collation requirements require an increase from  
43186 the default 2 to 4; no higher numbers are anticipated.

43187 {POSIX2\_EXPR\_NEST\_MAX}

43188 No increase is anticipated.

43189 {POSIX2\_LINE\_MAX}

43190 This number is much larger than most historical applications have been able to use. At  
43191 some future time, applications may be rewritten to take advantage of even larger  
43192 values, but this is unlikely at the present time.

43193 {POSIX2\_RE\_DUP\_MAX}

43194 No increase is anticipated.

43195 {POSIX2\_VERSION}

43196 This is actually not a limit, but a standard version stamp. Generally, a profile should  
43197 specify this volume of IEEE Std. 1003.1-200x by a name in the normative references  
43198 section, not this value.

### 43199 5.3.3 Optional Behavior

43200 In this volume of IEEE Std. 1003.1-200x, there are no instances of the terms unspecified,  
43201 undefined, or implementation-dependent, or the verbs “may” or “need not” that the developers  
43202 of this volume of IEEE Std. 1003.1-200x anticipate or sanction as suitable for profile or test  
43203 method citation. All of these are merely warnings to portable applications to avoid certain areas  
43204 that can vary from system to system, and even over time on the same system. In many cases,  
43205 these terms are used explicitly to allow for extensions, but profiles should not anticipate and  
43206 require such extensions; future versions of this volume of IEEE Std. 1003.1-200x may do so.

|    |                                         |      |                                             |          |
|----|-----------------------------------------|------|---------------------------------------------|----------|
| 2  | _POSIX_VDISABLE.....                    | 929  | crontab.....                                | 310      |
| 3  | actions equivalent to functions.....    | 13   | fg.....                                     | 476      |
| 4  | admin.....                              | 160  | jobs.....                                   | 555      |
| 5  | ADV.....                                | 14   | nice.....                                   | 698      |
| 6  | AIO.....                                | 14   | nohup.....                                  | 711      |
| 7  | alias.....                              | 165  | renice.....                                 | 856      |
| 8  | alias substitution.....                 | 40   | BAR.....                                    | 14       |
| 9  | AND lists.....                          | 75   | basename.....                               | 222      |
| 10 | AND-OR list.....                        | 73   | batch.....                                  | 225      |
| 11 | appending redirected output.....        | 61   | bc.....                                     | 228      |
| 12 | ar.....                                 | 168  | grammar.....                                | 229      |
| 13 | archives                                |      | lexical conventions.....                    | 231      |
| 14 | ar command.....                         | 168  | operations.....                             | 233      |
| 15 | ARG_MAX.....                            | 1106 | operators.....                              | 233      |
| 16 | arithmetic expansion.....               | 56   | bcc (mailer blind carbon copy).....         | 641      |
| 17 | arithmetic language                     |      | BC_BASE_MAX.....                            | 22       |
| 18 | bc.....                                 | 228  | BC_DIM_MAX.....                             | 22       |
| 19 | array identifiers.....                  | 233  | BC_SCALE_MAX.....                           | 22       |
| 20 | asa.....                                | 175  | BC_STRING_MAX.....                          | 22, 231  |
| 21 | asynchronous lists.....                 | 74   | BE.....                                     | 14       |
| 22 | at.....                                 | 178  | bg.....                                     | 243      |
| 23 | at-job.....                             | 178  | binary primaries.....                       | 947      |
| 24 | automatic storage class.....            | 237  | break.....                                  | 97       |
| 25 | awk.....                                | 187  | break special built-in.....                 | 97       |
| 26 | actions.....                            | 198  | builtin.....                                | 298      |
| 27 | arithmetic functions.....               | 200  | c89.....                                    | 246      |
| 28 | escape sequences.....                   | 195  | external symbols.....                       | 250      |
| 29 | expression patterns.....                | 198  | standard libraries.....                     | 249      |
| 30 | expressions.....                        | 190  | cal.....                                    | 257      |
| 31 | functions.....                          | 200  | can.....                                    | 7        |
| 32 | grammar.....                            | 204  | carriage-control characters.....            | 175      |
| 33 | input/output and general functions..... | 202  | case conditional construct.....             | 77       |
| 34 | lexical conventions.....                | 210  | cat.....                                    | 259      |
| 35 | output statements.....                  | 199  | cc (mailer carbon copy).....                | 641      |
| 36 | overall program structure.....          | 189  | CD.....                                     | 14       |
| 37 | pattern ranges.....                     | 198  | cd.....                                     | 263      |
| 38 | patterns.....                           | 197  | cflow.....                                  | 267      |
| 39 | regular expressions.....                | 195  | changing the current working directory..... | 13       |
| 40 | special patterns.....                   | 197  | character counting.....                     | 1088     |
| 41 | string functions.....                   | 201  | charmap                                     |          |
| 42 | user-defined functions.....             | 203  | with localedef.....                         | 591      |
| 43 | variables and special variables.....    | 193  | writing names with locale.....              | 586      |
| 44 | background work                         |      | charmap file.....                           | 590, 929 |
| 45 | at.....                                 | 178  | CHAR_BIT.....                               | 381      |
| 46 | batch.....                              | 225  | checksums                                   |          |
| 47 | bg.....                                 | 243  | cksum.....                                  | 283      |

|    |                                         |         |                                             |     |
|----|-----------------------------------------|---------|---------------------------------------------|-----|
| 48 | chgrp .....                             | 270     | cp.....                                     | 302 |
| 49 | CHILD_MAX.....                          | 74      | CPT .....                                   | 15  |
| 50 | chmod .....                             | 273     | CPU time .....                              | 957 |
| 51 | grammar.....                            | 276     | cron daemon .....                           | 313 |
| 52 | chown.....                              | 279     | crontab .....                               | 310 |
| 53 | cksum.....                              | 283     | CS .....                                    | 15  |
| 54 | cmp.....                                | 288     | csplit .....                                | 314 |
| 55 | codeset conversion.....                 | 539     | ctags.....                                  | 318 |
| 56 | tr.....                                 | 966     | current working directory .....             | 10  |
| 57 | COLL_WEIGHTS_MAX .....                  | 22      | cut .....                                   | 323 |
| 58 | colon.....                              | 99      | CX .....                                    | 15  |
| 59 | colon special built-in.....             | 99, 101 | cxref.....                                  | 327 |
| 60 | comm.....                               | 291     | data keywords.....                          | 783 |
| 61 | command.....                            | 294     | date .....                                  | 330 |
| 62 | command mode .....                      | 369     | field descriptors .....                     | 330 |
| 63 | command search and execution.....       | 69      | modified field descriptors .....            | 332 |
| 64 | command substitution .....              | 54      | dd .....                                    | 337 |
| 65 | communications commands                 |         | delta .....                                 | 344 |
| 66 | mailx.....                              | 619     | df .....                                    | 348 |
| 67 | talk.....                               | 939     | diff.....                                   | 352 |
| 68 | uucp .....                              | 1008    | default output format.....                  | 354 |
| 69 | uudecode.....                           | 1012    | directory comparison format.....            | 353 |
| 70 | uuencode.....                           | 1015    | –c or –C output format .....                | 355 |
| 71 | uustat .....                            | 1020    | –e output format.....                       | 354 |
| 72 | uux.....                                | 1023    | –f output format .....                      | 355 |
| 73 | write .....                             | 1098    | directory commands                          |     |
| 74 | compilers                               |         | cd.....                                     | 263 |
| 75 | c89.....                                | 246     | pwd .....                                   | 794 |
| 76 | fort77 .....                            | 497     | directory lister .....                      | 604 |
| 77 | yacc.....                               | 1108    | dirname.....                                | 359 |
| 78 | compound commands.....                  | 75      | disk space commands                         |     |
| 79 | compound-list .....                     | 73      | df .....                                    | 348 |
| 80 | compress.....                           | 299     | du .....                                    | 362 |
| 81 | compression                             |         | ulimit.....                                 | 981 |
| 82 | compress .....                          | 299     | documentation.....                          | 664 |
| 83 | uncompress .....                        | 993     | dot .....                                   | 103 |
| 84 | zcat .....                              | 1124    | dot special built-in.....                   | 103 |
| 85 | concurrent execution of processes ..... | 10      | double-quotes.....                          | 36  |
| 86 | configuration values .....              | 517     | du .....                                    | 362 |
| 87 | conforming application.....             | 35, 172 | duplicating an input file descriptor .....  | 62  |
| 88 | consequences of shell errors.....       | 65      | duplicating an output file descriptor ..... | 62  |
| 89 | continue.....                           | 101     | echo.....                                   | 366 |
| 90 | control characters .....                | 926     | ed.....                                     | 369 |
| 91 | controlling terminal .....              | 10      | addresses.....                              | 371 |
| 92 | Coordinated Universal Time (UTC).....   | 335     | append command .....                        | 373 |
| 93 | copy files commands                     |         | change command .....                        | 373 |
| 94 | cp.....                                 | 302     | commands.....                               | 372 |
| 95 | dd .....                                | 337     | copy command .....                          | 378 |
| 96 | ln .....                                | 582     | delete command .....                        | 374 |
| 97 | mv.....                                 | 689     | edit command .....                          | 374 |
| 98 | pax .....                               | 737     | edit without checking command .....         | 374 |

|     |                                              |      |                            |          |
|-----|----------------------------------------------|------|----------------------------|----------|
| 99  | filename command .....                       | 374  | autoprint option .....     | 426      |
| 100 | global command .....                         | 374  | autowrite option .....     | 426      |
| 101 | global non-matched command .....             | 379  | beautify option .....      | 426      |
| 102 | help command .....                           | 375  | change command .....       | 404      |
| 103 | help-mode command .....                      | 375  | chdir command .....        | 405      |
| 104 | insert command .....                         | 375  | command descriptions ..... | 401      |
| 105 | interactive global command .....             | 375  | copy command .....         | 405      |
| 106 | interactive global not-matched command ..... | 379  | delete command .....       | 405      |
| 107 | join command .....                           | 376  | directory option .....     | 426      |
| 108 | line number command .....                    | 379  | edcompatible option .....  | 427      |
| 109 | list command .....                           | 376  | edit command .....         | 405      |
| 110 | mark command .....                           | 376  | edit options .....         | 425      |
| 111 | move command .....                           | 376  | errorbells option .....    | 427      |
| 112 | null command .....                           | 380  | escape command .....       | 422      |
| 113 | number command .....                         | 376  | execute command .....      | 424      |
| 114 | print command .....                          | 377  | exrc command .....         | 427      |
| 115 | prompt command .....                         | 377  | file command .....         | 406      |
| 116 | quit command .....                           | 377  | global command .....       | 407      |
| 117 | quit without checking command .....          | 377  | ignorecase option .....    | 427      |
| 118 | read command .....                           | 377  | insert command .....       | 408      |
| 119 | regular expressions .....                    | 370  | join command .....         | 408      |
| 120 | shell escape command .....                   | 379  | lisp command .....         | 427      |
| 121 | substitute command .....                     | 377  | list command .....         | 409, 427 |
| 122 | undo command .....                           | 378  | magic command .....        | 428      |
| 123 | write command .....                          | 379  | map command .....          | 409      |
| 124 | editors .....                                |      | mark command .....         | 410      |
| 125 | ed .....                                     | 369  | mesg command .....         | 428      |
| 126 | ex .....                                     | 388  | move command .....         | 411      |
| 127 | sed .....                                    | 879  | next command .....         | 411      |
| 128 | vi .....                                     | 1030 | number command .....       | 412      |
| 129 | ED_FILE_MAX .....                            | 381  | number option .....        | 428      |
| 130 | ED_LINE_MAX .....                            | 381  | open command .....         | 412      |
| 131 | effective group ID .....                     | 10   | paragraphs option .....    | 428      |
| 132 | effective user ID .....                      | 10   | preserve command .....     | 412      |
| 133 | Eighth Edition UNIX .....                    | 298  | print command .....        | 413      |
| 134 | env .....                                    | 385  | prompt command .....       | 428      |
| 135 | EPERM .....                                  | 307  | put command .....          | 413      |
| 136 | escape character (backslash) .....           | 36   | quit command .....         | 413      |
| 137 | escape sequences .....                       |      | read command .....         | 414      |
| 138 | awk .....                                    | 195  | readonly command .....     | 429      |
| 139 | gencat .....                                 | 508  | recover command .....      | 414      |
| 140 | lex .....                                    | 573  | redraw command .....       | 429      |
| 141 | establish the locale .....                   | 13   | regular expressions .....  | 424      |
| 142 | eval .....                                   | 105  | remap command .....        | 429      |
| 143 | eval special built-in .....                  | 105  | replacement strings .....  | 425      |
| 144 | ex .....                                     | 388  | report command .....       | 429      |
| 145 | addressing .....                             | 394  | rewind command .....       | 415      |
| 146 | adjust window command .....                  | 421  | scroll command .....       | 430      |
| 147 | append command .....                         | 404  | sections command .....     | 430      |
| 148 | args command .....                           | 404  | set command .....          | 415      |
| 149 | autoindent option .....                      | 425  | shell command .....        | 415      |

|     |                                   |                                    |                                |        |
|-----|-----------------------------------|------------------------------------|--------------------------------|--------|
| 150 | shell option .....                | 430                                | fc .....                       | 470    |
| 151 | shift left command .....          | 423                                | FD .....                       | 15     |
| 152 | shift right command .....         | 423                                | fg .....                       | 476    |
| 153 | shiftwidth option .....           | 430                                | field splitting .....          | 58     |
| 154 | showmatch option .....            | 430                                | FIFO special files .....       | 674    |
| 155 | showmode command .....            | 430                                | file .....                     | 479    |
| 156 | slowopen command .....            | 431                                | file access permissions .....  | 10     |
| 157 | source command .....              | 416                                | file comparisons .....         |        |
| 158 | substitute command .....          | 416                                | cmp .....                      | 288    |
| 159 | suspend command .....             | 417                                | comm .....                     | 291    |
| 160 | tabstop option .....              | 431                                | diff .....                     | 352    |
| 161 | tag command .....                 | 417                                | uniq .....                     | 1002   |
| 162 | tags command .....                | 431                                | file contents .....            | 12     |
| 163 | term command .....                | 431                                | file conversion .....          |        |
| 164 | terse command .....               | 431                                | cut .....                      | 323    |
| 165 | unabbrev command .....            | 418                                | dd .....                       | 337    |
| 166 | undo command .....                | 418                                | expand .....                   | 460    |
| 167 | unmap command .....               | 418                                | fold .....                     | 494    |
| 168 | visual command .....              | 419                                | head .....                     | 536    |
| 169 | warn command .....                | 431                                | join .....                     | 559    |
| 170 | window command .....              | 432                                | od .....                       | 714    |
| 171 | wrapmargin option .....           | 432                                | paste .....                    | 722    |
| 172 | write command .....               | 419                                | patch .....                    | 726    |
| 173 | write line number command .....   | 424                                | sort .....                     | 906    |
| 174 | writeany option .....             | 433                                | strings .....                  | 916    |
| 175 | xit command .....                 | 420                                | tail .....                     | 935    |
| 176 | yank command .....                | 421                                | tr .....                       | 966    |
| 177 | exec .....                        | 107                                | tsort .....                    | 974    |
| 178 | Fam .....                         | 712                                | unexpand .....                 | 996    |
| 179 | exec family .....                 | 81, 298, 696, 1105-1106            | uniq .....                     | 1002   |
| 180 | exec special built-in .....       | 107                                | uudecode .....                 | 1012   |
| 181 | exit .....                        | 109                                | uuencode .....                 | 1015   |
| 182 | exit special built-in .....       | 109                                | file creation .....            | 11     |
| 183 | exit status and errors .....      | 65                                 | file descriptor .....          | 10, 60 |
| 184 | exit status for commands .....    | 65                                 | file mode creation mask .....  | 10     |
| 185 | expand .....                      | 460                                | file permission commands ..... |        |
| 186 | export .....                      | 111                                | chgrp .....                    | 270    |
| 187 | export special built-in .....     | 111                                | chmod .....                    | 273    |
| 188 | expr .....                        | 463                                | chown .....                    | 279    |
| 189 | matching expression .....         | 465                                | umask .....                    | 984    |
| 190 | string operand .....              | 465                                | file read .....                | 11     |
| 191 | expression argument .....         | 199                                | file removal .....             | 12     |
| 192 | expression list .....             | 199                                | file searching .....           |        |
| 193 | EXPR_NEST_MAX .....               | 22                                 | grep .....                     | 528    |
| 194 | extended regular expression ..... | 187, 195, 306                      | file time values .....         | 12     |
| 195 | .....                             | 489, 528, 572, 691, 746, 861, 1103 | file tree commands .....       |        |
| 196 | extension .....                   |                                    | diff .....                     | 352    |
| 197 | CX .....                          | 15                                 | find .....                     | 486    |
| 198 | OH .....                          | 17                                 | ls .....                       | 604    |
| 199 | XSI .....                         | 20                                 | mkdir .....                    | 671    |
| 200 | false .....                       | 468                                | rmdir .....                    | 868    |

## Index

|     |                                  |        |                               |                                  |
|-----|----------------------------------|--------|-------------------------------|----------------------------------|
| 201 | file write.....                  | 11     | iconv.....                    | 539                              |
| 202 | filters                          |        | id.....                       | 542                              |
| 203 | asa.....                         | 175    | if conditional construct..... | 78                               |
| 204 | awk.....                         | 187    | implementation-dependent..... | 7                                |
| 205 | compress.....                    | 299    | inference rule.....           | 643                              |
| 206 | dd.....                          | 337    | input field separator.....    | 891                              |
| 207 | expand.....                      | 460    | input mode.....               | 369                              |
| 208 | fold.....                        | 494    | IP6.....                      | 15                               |
| 209 | head.....                        | 536    | ipcrm.....                    | 5, 546                           |
| 210 | iconv.....                       | 539    | ipcs.....                     | 5, 548                           |
| 211 | more.....                        | 677    | jobs.....                     | 555                              |
| 212 | nl.....                          | 702    | join.....                     | 559                              |
| 213 | paste.....                       | 722    | kill.....                     | 564                              |
| 214 | pax.....                         | 737    | legacy.....                   | 7                                |
| 215 | pr.....                          | 772    | lex.....                      | 568                              |
| 216 | read.....                        | 853    | actions.....                  | 574                              |
| 217 | sed.....                         | 879    | definitions.....              | 570                              |
| 218 | tail.....                        | 935    | escape sequences.....         | 573                              |
| 219 | tee.....                         | 943    | regular expressions.....      | 572                              |
| 220 | tr.....                          | 966    | rules.....                    | 571                              |
| 221 | uncompress.....                  | 993    | table sizes.....              | 571                              |
| 222 | unexpand.....                    | 996    | user subroutines.....         | 572                              |
| 223 | zcat.....                        | 1124   | lex, translation table.....   | 578                              |
| 224 | find.....                        | 486    | libraries                     |                                  |
| 225 | fold.....                        | 494    | ar command.....               | 168                              |
| 226 | for loop.....                    | 76     | LIMIT.....                    | 21                               |
| 227 | fort77.....                      | 497    | line counting.....            | 1088                             |
| 228 | external symbols.....            | 500    | LINE_MAX.....                 | 22, 188, 381-382, 640, 887, 1005 |
| 229 | standard libraries.....          | 499    | link.....                     | 5, 580                           |
| 230 | FR.....                          | 15     | lists.....                    | 73                               |
| 231 | FSC.....                         | 15     | AND-OR.....                   | 73                               |
| 232 | function definition command..... | 79     | compound-list.....            | 73                               |
| 233 | function identifiers.....        | 233    | ln.....                       | 582                              |
| 234 | fuser.....                       | 5, 503 | locale.....                   | 586                              |
| 235 | g-file.....                      | 344    | localedef.....                | 591                              |
| 236 | gencat.....                      | 506    | locking file.....             | 277                              |
| 237 | escape sequences.....            | 508    | logger.....                   | 595                              |
| 238 | generated file.....              | 344    | logname.....                  | 597                              |
| 239 | get.....                         | 510    | lp.....                       | 599                              |
| 240 | getconf.....                     | 517    | LR(1) grammars.....           | 1122                             |
| 241 | getopts.....                     | 523    | ls.....                       | 604                              |
| 242 | global storage class.....        | 237    | m4.....                       | 612                              |
| 243 | GNU make.....                    | 659    | macro processor.....          | 612                              |
| 244 | grep.....                        | 528    | mailx.....                    | 619                              |
| 245 | grouping commands.....           | 75     | change current directory..... | 630                              |
| 246 | hash.....                        | 533    | change folder.....            | 631                              |
| 247 | head.....                        | 536    | command escapes.....          | 637                              |
| 248 | here-document.....               | 61     | commands.....                 | 629                              |
| 249 | history command                  |        | copy messages.....            | 630                              |
| 250 | fc.....                          | 470    | declare aliases.....          | 629                              |
| 251 | hunk.....                        | 728    | declare alternatives.....     | 630                              |

|     |                                       |          |                                          |               |
|-----|---------------------------------------|----------|------------------------------------------|---------------|
| 252 | delete aliases.....                   | 636      | MAN.....                                 | 15            |
| 253 | delete messages .....                 | 630      | man.....                                 | 664           |
| 254 | delete messages and display.....      | 631      | may .....                                | 7             |
| 255 | direct messages to mbox.....          | 633      | mesg.....                                | 668           |
| 256 | discard header fields.....            | 630      | message catalog generation .....         | 506           |
| 257 | display beginning of messages.....    | 636      | MF.....                                  | 16            |
| 258 | display current message number.....   | 637      | MIL-STD-1753 .....                       | 501           |
| 259 | display header summary .....          | 632      | mkdir.....                               | 671           |
| 260 | display list of folders .....         | 632      | mkfifo.....                              | 674           |
| 261 | display message.....                  | 634      | ML.....                                  | 16            |
| 262 | display message size.....             | 635      | MLR.....                                 | 16            |
| 263 | echo a string.....                    | 631      | MON.....                                 | 16            |
| 264 | edit message.....                     | 631, 636 | more.....                                | 677           |
| 265 | execute commands conditionally .....  | 633      | discard and refresh.....                 | 684           |
| 266 | exit .....                            | 631      | display position .....                   | 686           |
| 267 | follow up specified messages .....    | 632      | examine new file.....                    | 685           |
| 268 | help.....                             | 632      | examine next file.....                   | 685           |
| 269 | hold messages.....                    | 632      | examine previous file .....              | 685           |
| 270 | internal variables.....               | 626      | go to beginning of file.....             | 683           |
| 271 | invoke a shell.....                   | 635      | go to end-of-file .....                  | 683           |
| 272 | invoke shell command .....            | 637      | go to tag.....                           | 685           |
| 273 | list available commands .....         | 633      | help.....                                | 682           |
| 274 | mail a message.....                   | 633      | invoke editor .....                      | 685           |
| 275 | null command.....                     | 637      | mark position .....                      | 684           |
| 276 | pipe message .....                    | 633      | quit.....                                | 686           |
| 277 | process next specified message.....   | 633      | refresh the screen.....                  | 683           |
| 278 | quit.....                             | 634      | repeat search.....                       | 685           |
| 279 | read mailx commands from a file ..... | 635      | repeat search in reverse.....            | 685           |
| 280 | receive mode .....                    | 619      | return to mark .....                     | 684           |
| 281 | reply to a message.....               | 634      | return to previous position .....        | 684           |
| 282 | reply to a message list .....         | 634      | scroll backward one half screenful ..... | 683           |
| 283 | retain header fields.....             | 635      | scroll backward one line .....           | 683           |
| 284 | save messages.....                    | 635      | scroll forward one half screenful .....  | 683           |
| 285 | scroll header display.....            | 637      | scroll forward one line .....            | 682           |
| 286 | send mode.....                        | 619      | search backward for pattern.....         | 684           |
| 287 | set variables .....                   | 635      | search forward for pattern .....         | 684           |
| 288 | start-up .....                        | 626      | skip forward one line.....               | 683           |
| 289 | touch messages.....                   | 636      | motion command .....                     | 894           |
| 290 | undelete messages .....               | 636      | MPR.....                                 | 16            |
| 291 | unset variables .....                 | 636      | MSG.....                                 | 16            |
| 292 | write messages to a file .....        | 636      | mv .....                                 | 689           |
| 293 | make.....                             | 643      | NAME_MAX.....                            | 172, 764      |
| 294 | default rules.....                    | 653      | newgrp.....                              | 694           |
| 295 | inference rules.....                  | 650      | NGROUPS_MAX .....                        | 697           |
| 296 | internal macros .....                 | 652      | nice.....                                | 698           |
| 297 | libraries.....                        | 651      | Ninth Edition UNIX .....                 | 240, 368, 781 |
| 298 | macros.....                           | 649      | nl .....                                 | 702           |
| 299 | makefile execution.....               | 647      | nm.....                                  | 706           |
| 300 | makefile syntax.....                  | 646      | noclobber option.....                    | 63, 735       |
| 301 | target rules .....                    | 647      | nohup.....                               | 711           |
| 302 | make, GNU version .....               | 659      | non-printable .....                      | 381, 885, 942 |



|     |                                                     |     |                                                |                      |
|-----|-----------------------------------------------------|-----|------------------------------------------------|----------------------|
| 303 | OB .....                                            | 16  | OR lists .....                                 | 75                   |
| 304 | object files .....                                  | 706 | ordinary identifiers .....                     | 233                  |
| 305 | od .....                                            | 714 | paginators .....                               |                      |
| 306 | OF .....                                            | 16  | more .....                                     | 677                  |
| 307 | OH .....                                            | 17  | parameter expansion .....                      | 51                   |
| 308 | open file descriptors for reading and writing ..... | 63  | parameters and variables .....                 | 43                   |
| 309 | open mode .....                                     | 388 | paste .....                                    | 722                  |
| 310 | option .....                                        |     | patch .....                                    | 726                  |
| 311 | ADV .....                                           | 14  | file name determination .....                  | 729                  |
| 312 | AIO .....                                           | 14  | patch application .....                        | 729                  |
| 313 | BAR .....                                           | 14  | patchfile format .....                         | 728                  |
| 314 | BE .....                                            | 14  | path name expansion .....                      | 59                   |
| 315 | CD .....                                            | 14  | path name manipulation .....                   |                      |
| 316 | CPT .....                                           | 15  | basename .....                                 | 222                  |
| 317 | CS .....                                            | 15  | dirname .....                                  | 359                  |
| 318 | FD .....                                            | 15  | pathchk .....                                  | 732                  |
| 319 | FR .....                                            | 15  | path name resolution .....                     | 13                   |
| 320 | FSC .....                                           | 15  | pathchk .....                                  | 732                  |
| 321 | IP6 .....                                           | 15  | PATH_MAX .....                                 | 22, 863              |
| 322 | MF .....                                            | 16  | pattern matching .....                         | 486, 746, 1009, 1025 |
| 323 | ML .....                                            | 16  | definition .....                               | 92                   |
| 324 | MLR .....                                           | 16  | in case statements .....                       | 77                   |
| 325 | MON .....                                           | 16  | in shell variables .....                       | 52                   |
| 326 | MPR .....                                           | 16  | pattern matching notation .....                | 92, 491, 763         |
| 327 | MSG .....                                           | 16  | pattern scanning and processing language ..... |                      |
| 328 | PIO .....                                           | 17  | at .....                                       | 187                  |
| 329 | PS .....                                            | 17  | patterns matching a single character .....     | 92                   |
| 330 | RTS .....                                           | 17  | patterns matching multiple characters .....    | 93                   |
| 331 | RWL .....                                           | 17  | patterns used for file name expansion .....    | 94                   |
| 332 | SD .....                                            | 17  | pax .....                                      | 737                  |
| 333 | SEM .....                                           | 17  | PI .....                                       | 17                   |
| 334 | SHM .....                                           | 18  | PIO .....                                      | 17                   |
| 335 | SIO .....                                           | 18  | pipelines .....                                | 72                   |
| 336 | SPI .....                                           | 18  | positional parameters .....                    | 43                   |
| 337 | SPN .....                                           | 18  | POSIX2_BC_BASE_MAX .....                       | 21-22, 1134          |
| 338 | SS .....                                            | 18  | POSIX2_BC_DIM_MAX .....                        | 21-22, 1134          |
| 339 | TCT .....                                           | 18  | POSIX2_BC_SCALE_MAX .....                      | 21-22, 1134          |
| 340 | THR .....                                           | 18  | POSIX2_BC_STRING_MAX .....                     | 21-22, 1134          |
| 341 | TMO .....                                           | 19  | POSIX2_CHAR_TERM .....                         | 1134                 |
| 342 | TMR .....                                           | 19  | POSIX2_COLL_WEIGHTS_MAX .....                  | 21-22, 1134          |
| 343 | TPI .....                                           | 19  | POSIX2_C_BIND .....                            | 1133                 |
| 344 | TPP .....                                           | 19  | POSIX2_C_DEV .....                             | 1133                 |
| 345 | TPS .....                                           | 19  | POSIX2_EXPR_NEST_MAX .....                     | 21-22, 1134          |
| 346 | TSA .....                                           | 19  | POSIX2_FORT_DEV .....                          | 1133                 |
| 347 | TSF .....                                           | 19  | POSIX2_FORT_RUN .....                          | 1133                 |
| 348 | TSH .....                                           | 19  | POSIX2_LINE_MAX .....                          | 21, 23, 1134         |
| 349 | TSP .....                                           | 20  | POSIX2_LOCALEDEF .....                         | 1131, 1133           |
| 350 | TSS .....                                           | 20  | POSIX2_RE_DUP_MAX .....                        | 21, 23, 1134         |
| 351 | TYM .....                                           | 20  | POSIX2_SW_DEV .....                            | 1133                 |
| 352 | UP .....                                            | 20  | POSIX2_UPE .....                               | 1133                 |
| 353 | XSR .....                                           | 20  | , .....                                        | 1134                 |

|     |                                   |                                        |                                            |              |
|-----|-----------------------------------|----------------------------------------|--------------------------------------------|--------------|
| 354 | POSIX2_VERSION.....               | 21, 1134                               | rm.....                                    | <b>860</b>   |
| 355 | pr.....                           | <b>772</b>                             | rmel.....                                  | <b>865</b>   |
| 356 | print-related commands            |                                        | rmdir.....                                 | <b>868</b>   |
| 357 | fold.....                         | 494                                    | root directory.....                        | 10           |
| 358 | lp.....                           | 599                                    | RTS.....                                   | 17           |
| 359 | pr.....                           | 772                                    | RWL.....                                   | 17           |
| 360 | printf.....                       | 777                                    | sact.....                                  | <b>871</b>   |
| 361 | privileges.....                   | 669, 700                               | saved set-group-ID.....                    | 10           |
| 362 | process attributes.....           | 10                                     | saved set-user-ID.....                     | 10           |
| 363 | process group ID.....             | 10                                     | sccs.....                                  | <b>874</b>   |
| 364 | process ID.....                   | 10                                     | SCCS commands                              |              |
| 365 | process status report.....        | 787                                    | admin.....                                 | 160          |
| 366 | prompting.....                    | 48                                     | delta.....                                 | 344          |
| 367 | prs.....                          | <b>782</b>                             | get.....                                   | 510          |
| 368 | PS.....                           | 17                                     | prs.....                                   | 782          |
| 369 | ps.....                           | <b>787</b>                             | rmel.....                                  | 865          |
| 370 | public locale.....                | 586                                    | sact.....                                  | 871          |
| 371 | pwd.....                          | <b>794</b>                             | sccs.....                                  | 874          |
| 372 | qalter.....                       | <b>796</b>                             | unget.....                                 | 999          |
| 373 | qdel.....                         | <b>805</b>                             | val.....                                   | 1027         |
| 374 | qhold.....                        | <b>808</b>                             | what.....                                  | 1091         |
| 375 | qmove.....                        | <b>811</b>                             | SD.....                                    | 17           |
| 376 | qmsg.....                         | <b>814</b>                             | sed.....                                   | <b>879</b>   |
| 377 | qrerun.....                       | <b>817</b>                             | addresses.....                             | 881          |
| 378 | qrls.....                         | <b>819</b>                             | editing commands.....                      | 881          |
| 379 | qselect.....                      | <b>823</b>                             | regular expressions.....                   | 881          |
| 380 | qsig.....                         | <b>832</b>                             | SEM.....                                   | 17           |
| 381 | qstat.....                        | <b>835</b>                             | sequential lists.....                      | 75           |
| 382 | qsub.....                         | <b>840</b>                             | session membership.....                    | 10           |
| 383 | quote removal.....                | 59                                     | set.....                                   | 47, 117      |
| 384 | quoting.....                      | 36                                     | set special built-in.....                  | 117          |
| 385 | read.....                         | <b>853</b>                             | set-group-ID.....                          | 10, 308      |
| 386 | readonly.....                     | <b>113</b>                             | set-user-ID.....                           | 10, 278, 308 |
| 387 | readonly special built-in.....    | 113                                    | set-user-ID scripts.....                   | 901          |
| 388 | real group ID.....                | 10                                     | sh.....                                    | <b>888</b>   |
| 389 | real user ID.....                 | 10                                     | command history list.....                  | 892          |
| 390 | redirecting input.....            | 61                                     | command line editing.....                  | 892          |
| 391 | redirecting output.....           | 61                                     | vi line editing command mode.....          | 894          |
| 392 | redirection.....                  | 60                                     | vi line editing insert mode.....           | 893          |
| 393 | regular expressions.....          | 194-195, 306, 370                      | vi-mode command line editing.....          | 893          |
| 394 | .....                             | 394, 424, 465, 489, 528, 572, 682, 691 | shall.....                                 | 7            |
| 395 | .....                             | 702, 743, 861, 881, 1046, 1049, 1103   | shell command language.....                | 35           |
| 396 | related to shell patterns.....    | 92                                     | alias substitution.....                    | 40           |
| 397 | relational database operator..... | 559                                    | appending redirected output.....           | 61           |
| 398 | remove directories.....           | 868                                    | arithmetic expansion.....                  | 56           |
| 399 | remove files.....                 | 860                                    | command substitution.....                  | 54           |
| 400 | renice.....                       | <b>856</b>                             | compound commands.....                     | 75           |
| 401 | reserved words.....               | 42                                     | consequences of shell errors.....          | 65           |
| 402 | return.....                       | <b>115</b>                             | double-quotes.....                         | 36           |
| 403 | return special built-in.....      | 115                                    | duplicating an input file descriptor.....  | 62           |
| 404 | RE_DUP_MAX.....                   | 22                                     | duplicating an output file descriptor..... | 62           |

|     |                                                     |            |                                                        |                   |
|-----|-----------------------------------------------------|------------|--------------------------------------------------------|-------------------|
| 405 | escape character (backslash) .....                  | 36         | simple commands .....                                  | 67                |
| 406 | exit status and errors .....                        | 65         | single-quotes .....                                    | 36                |
| 407 | exit status for commands .....                      | 65         | SIO .....                                              | <b>18</b>         |
| 408 | field splitting .....                               | 58         | sleep .....                                            | <b>903</b>        |
| 409 | function definition command .....                   | 79         | SLR(1) grammars .....                                  | 1122              |
| 410 | grammar .....                                       | 82         | sort .....                                             | <b>906</b>        |
| 411 | here-document .....                                 | 61         | special built-in.80, 298, 700, 713, 795, 900, 958, 973 |                   |
| 412 | introduction .....                                  | 35         | special built-in utilities .....                       | 96                |
| 413 | lists .....                                         | 73         | break .....                                            | 97, 101           |
| 414 | open file descriptors for reading and writing       | 63         | characteristics .....                                  | 96                |
| 415 | parameter expansion .....                           | 51         | colon .....                                            | 99                |
| 416 | parameters and variables .....                      | 43         | dot .....                                              | 103               |
| 417 | path name expansion .....                           | 59         | eval .....                                             | 105               |
| 418 | pattern matching notation .....                     | 92         | exec .....                                             | 107               |
| 419 | patterns matching a single character .....          | 92         | exit .....                                             | 109               |
| 420 | patterns matching multiple characters .....         | 93         | export .....                                           | 111               |
| 421 | patterns used for file name expansion .....         | 94         | readonly .....                                         | 113               |
| 422 | pipelines .....                                     | 72         | return .....                                           | 115               |
| 423 | positional parameters .....                         | 43         | set .....                                              | 117               |
| 424 | quote removal .....                                 | 59         | shift .....                                            | 123               |
| 425 | quoting .....                                       | 36         | times .....                                            | 125               |
| 426 | redirecting input .....                             | 61         | trap .....                                             | 127               |
| 427 | redirecting output .....                            | 61         | unset .....                                            | 130               |
| 428 | redirection .....                                   | 60         | special parameters .....                               | 43                |
| 429 | reserved words .....                                | 42         | special targets .....                                  | 648               |
| 430 | shell commands .....                                | 67         | SPI .....                                              | <b>18</b>         |
| 431 | shell execution environment .....                   | 90         | split .....                                            | <b>912</b>        |
| 432 | shell grammar lexical conventions .....             | 82         | split files                                            |                   |
| 433 | shell grammar rules .....                           | 82         | csplit .....                                           | 314               |
| 434 | shell variables .....                               | 45         | split .....                                            | 912               |
| 435 | signals and error handling .....                    | 89         | SPN .....                                              | <b>18</b>         |
| 436 | simple commands .....                               | 67         | spoofing .....                                         | 114               |
| 437 | single-quotes .....                                 | 36         | SS .....                                               | <b>18</b>         |
| 438 | special built-in utilities .....                    | 96         | standard error .....                                   | 60                |
| 439 | special parameters .....                            | 43         | standard input .....                                   | 60                |
| 440 | tilde expansion .....                               | 50         | standard output .....                                  | 60                |
| 441 | token recognition .....                             | 39         | strings .....                                          | <b>916</b>        |
| 442 | word expansions .....                               | 49         | strip .....                                            | <b>919</b>        |
| 443 | shell commands .....                                | 67         | stty .....                                             | <b>921</b>        |
| 444 | shell execution environment ..44, 90, 166, 855, 986 |            | combination modes .....                                | 927               |
| 445 | shell grammar .....                                 | 82         | control modes .....                                    | 921               |
| 446 | shell grammar lexical conventions .....             | 82         | input modes .....                                      | 922               |
| 447 | shell grammar rules .....                           | 82         | local modes .....                                      | 925               |
| 448 | shell introduction .....                            | 35         | output modes .....                                     | 924               |
| 449 | shell variables .....                               | 45         | special control character assignments .....            | 926               |
| 450 | shift .....                                         | <b>123</b> | st_gid .....                                           | 173               |
| 451 | shift special built-in .....                        | 123        | st_mode .....                                          | 173               |
| 452 | SHM .....                                           | <b>18</b>  | st_mtime .....                                         | 173               |
| 453 | should .....                                        | 7          | st_size .....                                          | 173               |
| 454 | signal processes .....                              | 564        | st_uid .....                                           | 173               |
| 455 | signals and error handling .....                    | 89         | superuser .....                                        | 48, 474, 610, 762 |

|     |                                   |            |                               |                  |
|-----|-----------------------------------|------------|-------------------------------|------------------|
| 456 | supplementary group IDs.....      | 10         | unary primaries.....          | 947              |
| 457 | system configuration values ..... | 517        | uncompress.....               | <b>993</b>       |
| 458 | system name .....                 | 990        | undefined .....               | 7                |
| 459 | tabs.....                         | <b>931</b> | unexpand.....                 | <b>996</b>       |
| 460 | tag file creation.....            | 318        | unset.....                    | <b>999</b>       |
| 461 | tail .....                        | <b>935</b> | uniq.....                     | <b>1002</b>      |
| 462 | talk .....                        | <b>939</b> | unlink .....                  | 5, 1006          |
| 463 | tar format.....                   | 753        | unset .....                   | <b>130</b>       |
| 464 | target rule .....                 | 643        | unset special built-in.....   | 130              |
| 465 | TCT .....                         | <b>18</b>  | unspecified .....             | 8                |
| 466 | tee.....                          | <b>943</b> | until loop .....              | 79               |
| 467 | terminal characteristics          |            | UP.....                       | <b>20</b>        |
| 468 | stty.....                         | 921        | user identity                 |                  |
| 469 | tabs .....                        | 931        | id .....                      | 542              |
| 470 | tput .....                        | 963        | logname.....                  | 597              |
| 471 | tty.....                          | 977        | newgrp.....                   | 694              |
| 472 | terminate processes.....          | 564        | who.....                      | 1094             |
| 473 | terminology.....                  | 7          | utility option parsing.....   | 523              |
| 474 | test.....                         | <b>946</b> | uucp.....                     | <b>1008</b>      |
| 475 | THR .....                         | <b>18</b>  | Cmd.....                      | 1131             |
| 476 | tilde expansion .....             | 50         | uudecode.....                 | <b>1012</b>      |
| 477 | time.....                         | <b>955</b> | uuencode.....                 | <b>1015</b>      |
| 478 | times .....                       | <b>125</b> | uustat .....                  | <b>1020</b>      |
| 479 | times special built-in.....       | 125        | uux.....                      | <b>1023</b>      |
| 480 | TMO .....                         | <b>19</b>  | val .....                     | <b>1027</b>      |
| 481 | TMR .....                         | <b>19</b>  | Version 7 .....               | 35               |
| 482 | token recognition.....            | 39         | vi .....                      | <b>1030</b>      |
| 483 | touch.....                        | <b>959</b> | append.....                   | 1051             |
| 484 | TPI.....                          | <b>19</b>  | change.....                   | 1051             |
| 485 | TPP.....                          | <b>19</b>  | change to end-of-line.....    | 1052             |
| 486 | TPS.....                          | <b>19</b>  | clear and redisplay .....     | 1038             |
| 487 | tput .....                        | <b>963</b> | command descriptions.....     | 1031             |
| 488 | tr.....                           | <b>966</b> | delete.....                   | 1052             |
| 489 | trap.....                         | <b>127</b> | delete character.....         | 1062             |
| 490 | trap special built-in .....       | 127        | delete to end-of-line.....    | 1053             |
| 491 | trojan horse .....                | 610        | display information .....     | 1037             |
| 492 | true.....                         | <b>972</b> | enter ex mode.....            | 1059             |
| 493 | TSA .....                         | <b>19</b>  | execute.....                  | 1050             |
| 494 | TSF.....                          | <b>19</b>  | execute an ex command.....    | 1048             |
| 495 | TSH.....                          | <b>19</b>  | exit .....                    | 1064             |
| 496 | tsort.....                        | <b>974</b> | find character .....          | 1054             |
| 497 | TSP .....                         | <b>20</b>  | find regular expression ..... | 1046             |
| 498 | TSS .....                         | <b>20</b>  | insert .....                  | 1055             |
| 499 | tty.....                          | <b>977</b> | insert empty line.....        | 1057             |
| 500 | TYM.....                          | <b>20</b>  | join.....                     | 1055             |
| 501 | type.....                         | <b>979</b> | mark position .....           | 1056             |
| 502 | ulimit.....                       | <b>981</b> | move back .....               | 1045, 1051       |
| 503 | umask.....                        | <b>984</b> | move cursor.....              | 1037, 1040, 1060 |
| 504 | UN.....                           | <b>20</b>  | move forward .....            | 1045             |
| 505 | unalias.....                      | <b>988</b> | move to bigword.....          | 1053, 1061       |
| 506 | uname .....                       | <b>990</b> | move to bottom of screen..... | 1056             |

|     |                                       |            |                                         |      |
|-----|---------------------------------------|------------|-----------------------------------------|------|
| 507 | move to first character in line ..... | 1048       | xargs .....                             | 1101 |
| 508 | move to first non-<blank> .....       | 1044       | XSI .....                               | 20   |
| 509 | move to line .....                    | 1054       | XSR .....                               | 20   |
| 510 | move to matching character .....      | 1041       | yacc .....                              | 1108 |
| 511 | move to middle of screen .....        | 1056       | algorithms .....                        | 1119 |
| 512 | move to next section .....            | 1043       | code file .....                         | 1110 |
| 513 | move to specific column .....         | 1045       | completing the program .....            | 1119 |
| 514 | move to top of screen .....           | 1055       | conflicts .....                         | 1116 |
| 515 | move to word .....                    | 1053, 1061 | debugging the parser .....              | 1119 |
| 516 | page backwards .....                  | 1035       | declarations section .....              | 1111 |
| 517 | page forward .....                    | 1036       | description file .....                  | 1110 |
| 518 | put from buffer .....                 | 1057-1058  | error handling .....                    | 1117 |
| 519 | redraw screen .....                   | 1038       | grammar rules .....                     | 1112 |
| 520 | regular expression .....              | 1049       | header file .....                       | 1110 |
| 521 | reindent .....                        | 1050       | input grammar .....                     | 1114 |
| 522 | repeat .....                          | 1046       | input language .....                    | 1110 |
| 523 | repeat find .....                     | 1049, 1057 | interface to the lexical analyzer ..... | 1118 |
| 524 | repeat substitution .....             | 1042       | lexical structure of the grammar .....  | 1111 |
| 525 | replace character .....               | 1059       | library .....                           | 1119 |
| 526 | replace text with command .....       | 1040       | limits .....                            | 1120 |
| 527 | return to previous context .....      | 1042       | programs section .....                  | 1114 |
| 528 | return to previous section .....      | 1043       | zcat .....                              | 1124 |
| 529 | reverse case .....                    | 1050       |                                         |      |
| 530 | reverse find character .....          | 1046       |                                         |      |
| 531 | scroll backward .....                 | 1038       |                                         |      |
| 532 | scroll forward .....                  | 1036       |                                         |      |
| 533 | scroll forward by line .....          | 1036       |                                         |      |
| 534 | shift left .....                      | 1049       |                                         |      |
| 535 | shift right .....                     | 1049       |                                         |      |
| 536 | substitute character .....            | 1059       |                                         |      |
| 537 | substitute lines .....                | 1060       |                                         |      |
| 538 | undo .....                            | 1060       |                                         |      |
| 539 | undo current line .....               | 1061       |                                         |      |
| 540 | yank .....                            | 1062       |                                         |      |
| 541 | yank current line .....               | 1063       |                                         |      |
| 542 | visual mode .....                     | 388        |                                         |      |
| 543 | wait .....                            | 1084       |                                         |      |
| 544 | warning                               |            |                                         |      |
| 545 | MAN .....                             | 15         |                                         |      |
| 546 | OB .....                              | 16         |                                         |      |
| 547 | OF .....                              | 16         |                                         |      |
| 548 | PI .....                              | 17         |                                         |      |
| 549 | UN .....                              | 20         |                                         |      |
| 550 | wc .....                              | 1088       |                                         |      |
| 551 | what .....                            | 1091       |                                         |      |
| 552 | while loop .....                      | 78         |                                         |      |
| 553 | who .....                             | 1094       |                                         |      |
| 554 | will .....                            | 8          |                                         |      |
| 555 | word counting .....                   | 1088       |                                         |      |
| 556 | word expansions .....                 | 49         |                                         |      |
| 557 | write .....                           | 1098       |                                         |      |

